Unsupervised Anomaly Detection for Credit Card Fraud

Marta Rodriguez

May 2021

Abstract

Anomalous detection unsupervised algorithms, in detail, Mahalanobis distance, Local Outlier Factor, Density based spatial clustering of applications with noise and Gaussian Mixture Models has been discussed in this study. The suitability of these algorithms to detect anomalies in a credit card data base are the Gordian knot of this project, measured through a series of metrics.

Contents

Chapter 1

Introduction

Detecting anomalies has been of great interest over the past few years (Ghorbani, 2019; Goldstein and Uchida, 2016). The motivation for detecting anomalies back then was to remove them as the pattern recognition methods were sensitive to this observations (Goldstein and Uchida, 2016). However in the year 2000, the anomalies in themselves started becoming more interesting for researchers, since they are used in many applications such as network intrusion detection, fraud detection and medical applications (Goldstein and Uchida, 2016).

An outlier is a point that differs significantly from the rest of the data (Aggarwal, 2015; Ghorbani, 2019). Hawkins (1980) defines an outlier as follows: "an outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". Similarly, Goldstein and Uchida (2016); Mehrotra et al. (2017) specify anomalies as observations that deviate from the norm. As anomalies occur less often than non-anomalous points they can be considered outliers (Mehrotra et al., 2017). In this project, we will consider them as outliers because in our application frauds tend to be rare in big data sets.

These anomalous observations can be classified as global anomalies which are instances that differ from dense areas of the data set and local anomalies that differ with their nearest neighbours (Ester et al., 1996; Goldstein and Uchida, 2016). Our interest is mainly in the detecting the latter anomalies because in the real world the structures of the data sets are complex and there will be anomalies between sets of normal points. In addition, by detecting local anomalies we are indirectly detecting the global ones, as they will also deviate from their neighbouring points.

Anomalies can also be classified into point anomalies, collective anomalies and contextual anomalies (Goldstein and Uchida, 2016):

- **Point anomalies:** single anomalous observations.
- **Collective anomalies:** sets of anomalous observations
- **Contextual anomalies:** observations that are anomalous with respect to a given context

The main focus of this project will be to detect the first two: point anomalies and collective anomalies.

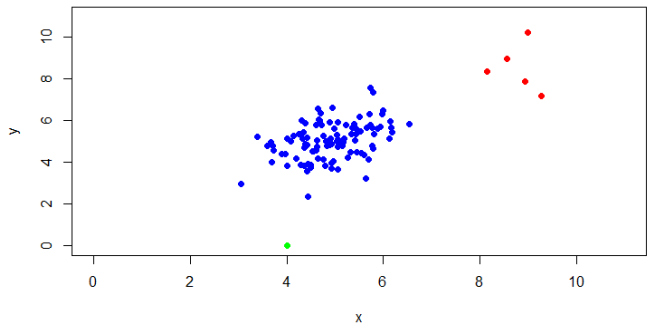

Figure 1.1: Plot where the red points are the collective anomalies, the green point is the point anomaly and the blue points are the normal points

Anomalies can be detected through anomaly detection methods (Goldstein and Uchida, 2016). Depending on the data available, the methods can be classified as follows (Goldstein and Uchida, 2016; Mehrotra et al., 2017):

- **Supervised Anomaly Detection:** the methods use labelled data to construct detection models to predict labels for future observations.
- **Semi-supervised Anomaly Detection:** only part of the data is labelled and also is based on the construction of models.
- **Unsupervised Anomaly Detection:** the labels of the data are unknown and detect anomalies based on the structure and properties of the data.

Unsupervised Anomaly Detection methods are more flexible than supervised methods as it does not require labels (Goldstein and Uchida, 2016). Furthermore, new type anomalies can occur after the construction of the model and it may no longer fit the new data, leading to to wrong predictions (Goldstein and Uchida, 2016). Hence, it is only able to detect certain anomalies that are similar to the ones that have already occurred (Mehrotra et al., 2017). However the data is not expected to follow previous patters (Mehrotra et al., 2017). In this project the focus will be on unsupervised anomaly detection methods.

There are different anomaly detection approaches and can be classified as distance-based, density-based and rank-based methods as follows (Mehrotra et al., 2017):

- **Distance-based methods:** these rely on distance metrics so that the points further away from the rest of the data are considered anomalous
- **Density-based methods:** identify anomalies as observations lying in low density regions
- **Rank-bases:** methods where anomalies are identified as those points whose nearest neighbours have other observations as nearest neighbours

These methods either assign each observation a label identifying the point as anomalous or normal or a score representing the degree of abnormality of the point (Goldstein and Uchida, 2016).

Clustering can play an important role in anomaly detection (Mehrotra et al., 2017). A cluster is a collection of observations that are close to each other (Mehrotra et al., 2017). Sometimes it is useful to consider the data as a combination of clusters of observations (Mehrotra et al., 2017). Points distant from their neighbouring clusters can be considered anomalous, the most effective methods in detecting these anomalies are the density-clustering algoruthms (Mehrotra et al., 2017). Often in data sets small clusters of anomalies occur as they might have been created by the same underlying process, like a type of credit fraud (Aggarwal, 2015; Goldstein and Uchida, 2016).

Chapter 2

Methods

In this chapter four methods will be analyzed and explained four unsupervised methods, a distance-based method which is the Mahalanobis distance and three density-based methods, the LOF algorithm, the DBSCAN algorithm and the Gaussian mixture model. These will then be applied to a specific data set.

This data set can be summarized in a feature matrix $X \in \mathbb{R}^{n \times q}$. It is a matrix that has n rows, representing the sample size and q columns, which represents the number of attributes each observation has. It is defined as follows:

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mm} \end{pmatrix} \tag{2.1}$$

In X there can be latent samples which can have different means and/or covariances matrices and might have been generated by a different process. Detecting these types of samples is the main focus of anomaly detection methods, as these observations constitute the anomalies. And some of these anomalies can have interesting meanings in certain applications.

2.1 Mahalanobis distance

2.1.1 Introduction

The Mahalanobis distance was developed by P.C. Mahalanobis in 1930 when he was conducting his studies on racial likeness (McLachlan, 1999; Rencher, 2003). From that moment on, this metric has been very important in statistical analysis and has had many applications in classification, numerical taxonomy and pattern recognition (McLachlan, 1999). It is applied in many fields such as finance (Ghorbani, 2019; Stöckl and Hanke, 2014) but also in image processing, neurocomputing and physics (Ghorbani, 2019). One of the main applications of Mahalanobis distance is outlier detection (Ghorbani, 2019; De Maesschalck et al., 2000; Stöckl and Hanke, 2014).

The Mahalanobis distance, $\triangle$, of a data point $x \in \mathbb{R}^q$ to its population mean $\mu \in \mathbb{R}^q$

with covariance matrix $\Sigma \in \mathbb{R}^q \times \mathbb{R}^q$ is defined as follows (Ghorbani, 2019; Hadi, 1992; Leys et al., 2018; De Maesschalck et al., 2000; McLachlan, 1999):

$$\triangle(x, \mu, \Sigma) = \sqrt{(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{2.2}$$

The $\triangle$ is a metric since Σ is invertible as it is positive definite (McLachlan, 1999). The Mahalanobis distance increases as $(x-\mu)\Sigma^{-1}(x-\mu)^T$ increases (Ghorbani, 2019). In most cases, μ and Σ will be unknown so they need to be estimated by their respective unbiased estimators $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, the sample mean, and $S = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$, the sample covariance matrix. Then the Mahalanobis distance of a point to its mean is (Hadi, 1992; Leys et al., 2018; De Maesschalck et al., 2000; McLachlan, 1999; Penny, 1996):

$$d_M(x, \bar{x}, S) = \sqrt{(x-\bar{x})^T S^{-1} (x-\bar{x})} \tag{2.3}$$

The Mahalanobis distance d_M with the estimated parameters for the mean and covariance matrix is known to overestimate the true Mahalanobis distance $\triangle$ (McLachlan, 1999). It is a measure of similarity and even though there are many others, the Mahalanobis distance has been found to be the most appropriate for most applications (McLachlan, 1999). The Euclidean distance is another metric that could be employed. For a point $x \in \mathbb{R}^q$ to its population mean $\mu \in \mathbb{R}^q$, the Euclidean distance is as follows (De Maesschalck et al., 2000; Leys et al., 2018; Rencher, 2003):

$$D_E = \sqrt{(x_i - \mu)(x_i - \mu)^T} \tag{2.4}$$

However in multivariate space an appropriate distance metric needs to take into account the variances of the variables and their correlations, and the Euclidean distance does not (Rencher, 2003). The Euclidean distance assumes that each attribute in the data has the same importance and is independent of all other attributes (Xiang et al., 2008). However, this is not always the case in real applications, particularly in high dimensional spaces (Xiang et al., 2008). So it gives the same weight to highly correlated variables and less correlated variables (Ghorbani, 2019). Hence, as correlated variables account for basically the same characteristic, by assigning equal weight to correlated variables, the same variable gets counted more than once (Ghorbani, 2019).

On the other hand, Mahalanobis distance is able to solve this problem as it allows for unequal variances and correlations between features (Aggarwal, 2015; Xiang et al., 2008). Hence, if a variable has larger variance than another, it is given relatively less weight than in the Euclidean distance case (Rencher, 2003; Xiang et al., 2008). Furthermore, variables that are highly correlated are given less weight than variables less correlated in the Mahalanobis method (Rencher, 2003; Xiang et al., 2008). It can be seen that the inverse covariance matrix in the Mahalanobis distance is to take into consideration the scales of the variables and the correlations between variables (De Maesschalck et al., 2000; McLachlan, 1999; Rencher, 2003). Hence the Mahalanobis distance considers the correlations between features and it is scale invariant (Ghorbani, 2019).

The Mahalanobis distance transforms the variables into uncorrelated standardised variables and then computes the Euclidean distance of this transformed variables

(Aggarwal, 2015; Ghorbani, 2019). The Mahalanobis distance and the Euclidean distance will be equal if the covariance matrix is the identity matrix (McLachlan, 1999; Ghorbani, 2019). So this will be the case if each pair of variables were mutually uncorrelated and were scaled to have variance one (McLachlan, 1999; Xiang et al., 2008).

Nevertheless, computing these covariance matrix can be a problem if the data has too many variables as there can be a problem of multicollinearity that can result in the covariance matrix not being invertible and also to be able to compute the covariance matrix the size of the data set needs to be larger than the number of variables (De Maesschalck et al., 2000).

2.1.2 Mahalanobis distance for outlier detection

The Mahalanobis distance is a standard method for detecting outliers in multivariate data (Ghorbani, 2019; Penny, 1996). The Mahalanobis metric determines how close are two points from each other (Ghorbani, 2019). And as outliers differ from normal points this distance can be used to detect outliers (Ghorbani, 2019). As this method accounts for the structure of the data (Leys et al., 2018), it stands out from other distance metrics in its faculty to detect these outliers (Ghorbani, 2019). Aggarwal (2015); Ghorbani (2019); Krzanowski (2000); McLachlan (1999); Penny (1996) stated the following theorem that Krzanowski (2000) proved and is as follows:

Theorem 2.1.1. *If $X \sim N_q(\mu, \Sigma)$, then the squared Mahalanobis distance $\triangle^2(X, \mu, \Sigma) \sim \chi^2_q$. Where $N_q(\mu, \Sigma)$ is the q-variate normal distribution with mean μ and covariance matrix Σ and χ^2_q is the chi-squared distribution with q degrees of freedom.*

Proof.

$$\begin{aligned}
\triangle^2 &= (X-\mu)^T\Sigma^{-1}(X-\mu) \\
&= (X-\mu)^T(\Sigma^{1/2}\Sigma^{1/2})^{-1}(X-\mu) \\
&= (X-\mu)^T(\Sigma^{-1/2})^T\Sigma^{-1/2}(X-\mu) \\
&= [\Sigma^{-1/2}(X-\mu)]^T\Sigma^{-1/2}(X-\mu) \\
&= Z^TZ \\
&= \sum_{i=1}^{q} Z_i^2 \sim \chi^2_q
\end{aligned}$$

The sum of squares of q independent variables follows a χ^2-distribution with q degrees of freedom (Aggarwal, 2015). □

Hence the values $d_M^2 = (x-\bar{x})S^{-1}(x-\bar{x})^T$ can be used to test if the point x is an outlier (McLachlan, 1999). The most common method of Mahalanobis distance for outlier detection, estimates the mean μ and covariance matrix Σ by its unbiased estimators $\bar{x}$ and S and identifies as an outlier any point whose squared Mahalanobis distance is greater than a predetermined quantile of the chi-squared distribution of q degrees of freedom (Aggarwal, 2015; Ghorbani, 2019; Healy, 1968). So a hypothesis test can be established where the null hypothesis is $H_0 : x$ is an outlier and the alternative hypothesis is $H_1 : x$ is not an outlier (Aggarwal, 2015). Then the null hypothesis H_0

will be rejected at significance level α when $d_M^2(x, \bar{x}, S) > \chi_{q,\alpha}^2$ (Aggarwal, 2015; Becker and Gather, 1999; Healy, 1968). Then the method will be as follows (Aggarwal, 2015; Becker and Gather, 1999; Healy, 1968; McLachlan, 1999):

1. Set a significance level $\alpha \in (0, 1)$ and find the critical value $\chi_{q,\alpha}^2$.
2. Compute $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $S = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$.
3. For $i = 1, .., n$ calculate $d_M^2(x_i, \bar{x}, S)$.
   (a) If $d_M^2(x_i, \bar{x}, S) > \chi_{q,\alpha}^2$ label x_i as an anomalous point.
   (b) Else label x_i as a normal point.

However this method can cause problems as it relies on a multivariate assumption which is not always satisfied and the parameter estimates are sensitive to outlying points (Becker and Gather, 1999; Ghorbani, 2019), which can result in the misclassification of points. Two further problems arise form applying this method, the masking effect and the swamping problem (Hadi, 1992). The masking effect can occur when detecting multiple outliers, as one might mask another outlier by having an effect on the value of the mean, $\bar{x}$ and covariance matrix, S (Becker and Gather, 1999; Ghorbani, 2019; Hadi, 1992; Leys et al., 2018). Hence this masked outlier will be erroneously detected as a normal point (Hadi, 1992; Leys et al., 2018). The classical Mahalanobis method can have a high masking effect as the estimators of the mean and covariance matrix are not robust in the presence of outliers (Becker and Gather, 1999; Leys et al., 2018; De Maesschalck et al., 2000). The swamping problem occurs when some normal points are detected as anomalies due the influence of some outliers in the values of the sample mean, $\bar{x}$ and sample covariance matrix, S (Hadi, 1992). The cause of both of these problems is that the estimators are not robust to anomalies (De Maesschalck et al., 2000; Hadi, 1992; Leys et al., 2018).

For the reasons set out above, more robust estimators are needed to enhance the performance of the method (Becker and Gather, 1999; Ghorbani, 2019; Hadi, 1992; Leys et al., 2018). One way to obtain more robust estimators is by computing the Mahalanobis distance of each point x_i by deleting the i^{th} observation when estimating the mean and covariance matrix (Becker and Gather, 1999; De Maesschalck et al., 2000). This method is called the "jack-knifed" (Becker and Gather, 1999) or "leave-one-out" (De Maesschalck et al., 2000) Mahalanobis distance. So the new Mahalanobis distance for a point x_i is defined as follows (McLachlan, 1999):

$$d_{M(i)}(x_i, \bar{x}_{(i)}, S_{(i)}) = \sqrt{(x_i - \bar{x}_{(i)})^T S_{(i)}^{-1}(x_i - \bar{x}_{(i)})} \tag{2.5}$$

Here $\bar{x}_{(i)}$ is the sample mean and $\hat{\Sigma}_{(i)}$ is the sample covariance matrix when removing the point x_i. The method is then defined as follows (Becker and Gather, 1999; De Maesschalck et al., 2000; McLachlan, 1999):

1. Set a significance level $\alpha \in (0, 1)$ and find the critical value $\chi_{q,\alpha}^2$
2. For i=1,..,n

(a) Compute the estimators $\bar{x}_{(i)}$ and $S_{(i)}$, which are the sample mean, $\bar{x}$ and covariance matrix, S but deleting the i^{th} observation.

(b) Calculate $d_{M(i)}(x_i, \bar{x}_{(i)}, S_{(i)})$.

   i. If $d_{M(i)}(x_i, \bar{x}_{(i)}, S_{(i)}) > \chi^2_{q,\alpha}$ label x_i as an anomalous point.

   ii. Else label x_i as a normal point.

As a final note, anomalies can occur in small clusters as they can be caused by the same underlying process (like a specific type of credit fraud) (Aggarwal, 2015) and hence share similar characteristics. The Mahalanobis approach is able to detect this clusters as they often differ greatly from the global mean and covariance matrix of the data set (Aggarwal, 2015).

2.1.3 Advantages and disadvantages

"The method is robust to increasing dimensionality" as it accounts for the high dimensional variations in an effective way (Aggarwal, 2015). Although it is a simple method, it is able to take into consideration the dependencies between the variables in an effective way, which is especially important in high dimensional data sets (Aggarwal, 2015). A useful characteristic of the Mahalanobis distance is that it is invariant under affine transformations, hence variations in the scale of the variables is insignificant (Stöckl and Hanke, 2014).

This method outperforms other more complex distance-based approaches in computational complexity, accuracy and parametrization (Aggarwal, 2015). Major distributional assumptions are not needed for the Mahalanobis distance, it is an appropriate measure of distance for random variables in the multivariate space which are completely determined by the mean μ and covariance matrix Σ (Stöckl and Hanke, 2014). Observations with equal Mahalanobis distance are on an ellipsoid with position and shape determined by μ and Σ (Stöckl and Hanke, 2014).

In this approach, in principle, there is no need to estimate any parameters which can be very beneficial in unsupervised problems where the ground truth is unknown (Aggarwal, 2015).

Other distance-based methods have a computational complexity of $O(n^2)$, being n the size of the sample, this can be very costly for n large (Aggarwal, 2015). While Mahalanobis requires only $O(n)$ of time, however it needs quadratic time and space with regard to data dimensionality (Aggarwal, 2015). If the dimensionality of the data is high then the algorithm is computationally demanding (Xiang et al., 2008). However, since the sample size tends to be much higher than the number of dimensions, the Mahalanobis method is very efficient in most real-world data sets (Aggarwal, 2015). The Mahalanobis distance needs to go through every variable in the data to compute the inter-correlation structure, hence it is computationally more costly than the Euclidean distance (Ghorbani, 2019).

This simple method is able to detect small anomaly clusters while other more complex methods like Gaussian mixture models might miss it if the number of components is not chosen correctly (Aggarwal, 2015). However, if the outliers are located in sparse regions between clusters, this approach does not perform well (Aggarwal, 2015).

Aside from a method for detecting outliers, Mahalanobis distance can be used for other applications as it accounts for correlations between variables (Aggarwal, 2015). And can be combined with other methods to give robust results and outperform other outlier detectors (Aggarwal, 2015).

2.2 Local Outlier Factor Algorithm

2.2.1 Introduction and basic definitions

The Local Outlier Factor (LOF) method is an unsupervised algorithm proposed by Breunig et al. (2000). It is a density-based method (Kriegel et al., 2009; Lee et al., 2011), used to detect the outliers in a given set (Aggarwal, 2015; Duan et al., 2007; Lee et al., 2011). Instead of giving each point a binary label, the LOF method gives each point a local outlier score which represent the degree of isolation of an observation with respect to its surrounding neighbourhood (Aggarwal, 2015; Breunig et al., 1999, 2000; Duan et al., 2007; Kriegel et al., 2009; Lazarevic et al., 2003; Lee et al., 2011). This degree receives the name of Local Outlier Factor (LOF) of a point (Lazarevic et al., 2003). The only input parameter of the method is $MinPts \in \mathbb{N}$ which is the number of objects closest to a specific point used to define its local density and hence determine its LOF score (Breunig et al., 2000). Before developing the algorithm, there are some definitions that need to be introduced. The first definition is the k-distance of a specific point. Informally it is the distance from a point to its $MinPts^{th}$ closest point. In a more formal way (Aggarwal, 2015; Breunig et al., 1999, 2000; Goldstein and Uchida, 2016; Lazarevic et al., 2003; Lee et al., 2011; Mehrotra et al., 2017):

Definition 2.2.1. In a data set X, the $MinPts$-distance of a point p, $MinPts$-distance(p), is the distance between the points p and r, $d(p, r)$ that satisfies the following two conditions:

(i) There exist at least $MinPts$ points $y \in X \setminus \{p\}$ such that $d(p, y) \leq d(p, r)$.

(ii) There exists at most $MinPts - 1$ points $y \in X \setminus \{p\}$ such that $d(p, y) < d(p, r)$.

A concept related to $MinPts$-distance is the $MinPts$-distance neighborhood of a point p, which is the set of points that are less than or equal to the $MinPts$-distance(p). It is defined as follows (Aggarwal, 2015; Breunig et al., 1999, 2000; Goldstein and Uchida, 2016; Lazarevic et al., 2003; Lee et al., 2011; Mehrotra et al., 2017):

Definition 2.2.2. For a data set X and a point $p \in X$, the $MinPts$-distance neighbourhood of p, $N_{MinPts}(p)$ is defined as follows:

$$N_{MinPts}(p) = \{\ r \in X \mid d(p, r) \leq MinPts\text{−distance(p)}\} \tag{2.6}$$

As for a point p in the data set, this is called the $MinPts$-distance neighbourhood and by the definition of the $MinPts$-distance there are $MinPts$ objects expected to be in this neighbourhood. However, this is not always the case because there can be multiple points at a $MinPts$-distance from the point p, hence even though the $MinPts$-distance remains the same the cardinality of the neighbourhood can be greater than $MinPts$ (Breunig et al., 2000).

The LOF score needs to be defined taking into account that points in the same dense areas must have similar scores, which will make analyzing the outlying degree of the observations easier (Breunig et al., 2000). For that reason a new type of distance is introduced, called the reachibility-distance of a point p with respect to a point r and it is the maximum between the $MinPts$-distance of r and the distance between p and r (Breunig et al., 2000). In this way, points which are close to r will have equal reachibility distances from r and will lead to having similar outlier scores (Breunig et al., 2000). It can be defined as follows (Aggarwal, 2015; Breunig et al., 1999; Goldstein and Uchida, 2016; Lazarevic et al., 2003; Lee et al., 2011; Mehrotra et al., 2017):

Definition 2.2.3. For a set X, the reachibility-distance of a point p with respect to a point r is as follows:

$$reach\text{–}dist_{MinPts}(p, r) = \max\{MinPts\text{–}distance(r), d(p, r)\} \tag{2.7}$$

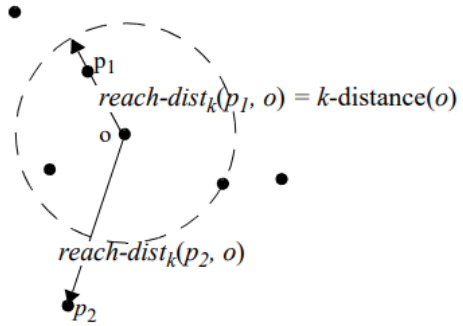

Figure 2.1: Graphical representation of the reachability distance of p_1 and p_2 with respect to o (Breunig et al., 2000)

Note that even though it is the reachibility distance of p with respect to r, the $MinPts$-distance of r is used to measure how approachable is p with respect to its $MinPts$-nearest neighbours. So if p is in the neighbourhood of r the reachibility distance with respect to r will be the $MinPts$-distance of r and otherwise it will be the actual distance (Breunig et al., 2000). This is done to compare the reachability distance of p with the reachibility distance of its $MinPts$-nearest neighbours, so if p is a neighbour of its nearest neighbours, then p will share similar reachibility distances with them and will not be flagged as an outlier (Breunig et al., 2000).

This algorithm needs to be in terms of densities, hence it is necessary to turn this reachibility distances of a point p with respect to its $MinPts$-nearest neighbours into some measure of density (Breunig et al., 2000). The larger the reachibility distances of p with respect to its neighbours, the further away is p from its neighbours and the smaller the local density of p should be, therefore the reachibility distance and the local density must have a negative correlation (Breunig et al., 2000).

This density measure introduced is the local reachibility density of p which is the inverse of the mean of the reachibility distances of p with respect to its neighbours (Breunig et al., 1999, 2000). This quantity can be infinite when the sum of the reachibility

distances of p with respect to its neighbours is 0, which can only happen if there is $MinPts$ duplicates of the point p, and this is a very unusual case even for $MinPts$ taking a small value and the data should be inspected further if this is the case before proceeding to further analysis (Breunig et al., 1999, 2000). Therefore, it can be assumed that no duplicates exist in the data set (Breunig et al., 1999, 2000). The definition of local reachibility distance is as follows (Aggarwal, 2015; Breunig et al., 1999; Goldstein and Uchida, 2016; Lazarevic et al., 2003; Lee et al., 2011; Mehrotra et al., 2017):

Definition 2.2.4. The local reachibility density, lrd_{MinPts}, of a point p:

$$lrd_{MinPts}(p) = 1/\left(\frac{\sum_{r \in N_{MinPts}(p)} reach - dist_{MinPts}(p,r)}{|N_{MinPts}(p)|}\right) \tag{2.8}$$

Once the local reachability density of a general point p in the data set is defined, it is necessary to compare this density with the local reachibility density of its neighbours to be able to give the point an outlier score, namely the Local Outlier Factor (Breunig et al., 2000). It is defined as follows (Aggarwal, 2015; Breunig et al., 1999; Goldstein and Uchida, 2016; Lazarevic et al., 2003; Lee et al., 2011; Mehrotra et al., 2017):

Definition 2.2.5. The local outlier factor, LOF_{MinPts}, of a point p:

$$LOF_{MinPts}(p) = \frac{\sum_{r \in N_{MinPts}(p)} \frac{lrd_{MinPts}(r)}{lrd_{MinPts}(p)}}{|N_{MinPts}(p)|} \tag{2.9}$$

The LOF method returns an outlier score for each point based on the local density (Aggarwal, 2015). The LOF score of a point is the average of the ratio of the local reachibility density of a point p and those of its $MinPts$-nearest neighbours (Duan et al., 2007; Breunig et al., 1999, 2000). The higher this ratio, the higher the LOF score of the point and the more distant it is with respect to the nearest cluster (Duan et al., 2007; Breunig et al., 2000). And, hence the possibility of the point being an outlier increases (Duan et al., 2007; Goldstein and Uchida, 2016; Mehrotra et al., 2017). If $LOF(p) < 1$ then the local reachibility density of p is greater than the one from its neighbours and hence the point is not an outlier (Goldstein and Uchida, 2016). If $LOF(p) \approx 1$ then the local reachability density of p will be similar to the one of its nearest neighbours and therefore it should not be classified as an outlier (Goldstein and Uchida, 2016). Breunig et al. (2000) showed that the majority of the points in a cluster have LOF scores close to one. On the other hand if $LOF(p) > 1$ then the density of p will be lower than the one of its $MinPts$-nearest neighbours (Duan et al., 2007). However, the question is how different this density has to be for a point p to be classified as an outlier, this question is much harder to answer and does not have a clear approach.

Hence, the interpretation of this factor to decide whether an observation is an outlier it is complex (Kriegel et al., 2009). And the threshold of LOF in which anomalies are distinguished from normal points is not clear (Goldstein and Uchida, 2016). Other variants of LOF, like the LoOP algorithm try to address this problem by assigning a probability of being an anomaly instead of a score (Goldstein and Uchida, 2016), in the range of [0,1], which is more easily interpretable (Kriegel et al., 2009).

2.2.2 Tuning the parameter $MinPts$

Breunig et al. (2000) defines the parameter $MinPts$ as "the number of nearest neighbours used in defining the local neighbourhood of the object". The LOF algorithm depends on the choice of $MinPts$ as the method is based on computing the distances of points to their $MinPts$-nearest neighbour (Kriegel et al., 2009). The greater the value of $MinPts$ the more analogous the reachibility-distances for points in the same region of the space (Breunig et al., 1999). Tuning the parameter $MinPts$ is essential for the LOF algorithm (Goldstein and Uchida, 2016). The performance of the LOF method is very sensitive the value of the parameter $MinPts$ and hence it needs to be carefully selected (Aggarwal, 2015; Kriegel et al., 2009). If the right value for $MinPts$ is chosen this method can outperform other distance-based methods (Aggarwal, 2015) like Mahalanobis distance and other density-based algorithms (Kriegel et al., 2009), however in unsupervised cases where the true labels are unknown, this tends to be a difficult task (Aggarwal, 2015). So the LOF method is very not robust when varying the value of the parameter $MinPts$ (Aggarwal, 2015; Kriegel et al., 2009) and this is a clear disadvantage of the method. The wrong choice of $MinPts$ can cause poor results (Kriegel et al., 2009). For example an unlucky choice of $MinPts$ can make the algorithm detect an outlier point as a normal point belonging to a cluster (Kriegel et al., 2009).

Given a range of $MinPts$ values, the LOF score tends to fluctuate and stabalize at some value (Breunig et al., 2000). As $MinPts$ takes greater values, the fluctuations in the reachability distances and in the LOF scores decrease (Breunig et al., 2000). To solve this problem Breunig et al. (2000) propose the method of taking a range of values of $MinPts$ and evaluate the LOF score at each of these values and set the maximum to be the LOF score for each point. For picking this range a lower bound, $MinPtsLB$, and an upperbound, $MinPtsUB$, must be chosen (Breunig et al., 2000). Eventhough Breunig et al. (2000) recommend using the LOF method by taking the maximum outlier score of each point for a range of values of $MinPts$, Goldstein and Uchida (2016) claim that the performance of this method and the method of taking single values of $MinPts$ is very similar.

The $MinPtsLB$ must be at least 1 as if it is zero, the neighbourhood of each point will just be itself and hence every reachibility distance will be infinite, which is hard to analyze. However if the value of $MinPts$ is too small the LOF scores of the points will (highly) fluctuate (Breunig et al., 2000), even for points in the same density areas (region of the space). This will make the task of distinguishing between normal points and anomalies very demanding. Hence, Breunig et al. (2000) recommend choosing the value of $MinPtsLB$ to be at least 10. Another guideline is that it is the minimum number of normal points a set (cluster) has to contain in order to be able to detect local outliers with respect to this set (cluster) (Breunig et al., 2000). Duan et al. (2007) argues that a value of MinPts between 10 and 20 tend to work well.

On the other hand, $MinPtsUB$ must be strictly less than $n-1$, where n is the sample size, as in this case every point will be in the same neighbourhood and no anomalies will be flagged. Nevertheless, if $MinPts$ is too large, the whole purpose of the algorithm detecting local outliers will be neglected, as some of the local outliers will be included in the nearest neighbour of normal points. So this upper bound can be identified as the maximum number of nearby points that can be local outliers (Breunig et al., 2000).

2.2.3 Method of LOF for detecting anomalies

Unlike other methods for detecting outliers like the Mahalanobis distance method, that take the whole dataset into account for deciding the outlierness of a point, the LOF method considers the $MinPts$ nearest neighbours of a point for determining its outlying degree (Kriegel et al., 2009). The focus of this method is local and not global (Kriegel et al., 2009). The method proposed by Breunig et al. (2000) is as follows:

1. Pick the value of $MinPtsLB$ and $MinPtsUB$ following the criterion defined in Section 2.
2. For i=1,...n
   (a) For $MinPtsLB \leq MinPts \leq MinPtsUB$
      i. Compute $MinPts$-distance(x_i) using the Euclidean metric.
      ii. Calculate the set $N_{MinPts}(x_i)$.
      iii. Find the $reach\text{–}dist_{MinPts}(x_i, o)$ for $\forall o \in N_{MinPts}(x_i)$.
      iv. Compute $lrd_{MinPts}(x_i) = 1/\left(\frac{\sum_{r \in N_{MinPts}(x_i)} reach - dist_{MinPts}(x_i, r)}{|N_{MinPts}(x_i)|}\right)$
      v. Find $LOF_{MinPts}(x_i) = \frac{\sum_{r \in N_{MinPts}(x_i)} \frac{lrd_{MinPts}(r)}{lrd_{MinPts}(x_i)}}{|N_{MinPts}(x_i)|}$
   (b) Set $LOF(x_i) = max\{LOF_{MinPts}(x_i) | MinPtsLB \leq MinPts \leq MinPtsUB\}$
3. Pick a threshold value for the LOF score c following the criteria explained at the end of Section 1.
   (a) If $LOF(x_i) > c$ label the point x_i as anomalous.
   (b) Else label x_i as non-anomalous.

LOF method is complementary with clustering techniques, while LOF finds outliers in sparse data, the clustering tecniques locate dense regions in the data (Aggarwal, 2015). For that reason clustering is also used in outlier analysis (Aggarwal, 2015). Clusters of anomalies are frequent because outliers tend to occur in small groups (Aggarwal, 2015). These clusters can also be identified by the proposed LOF method by setting the value of $MinPts$ higher than the number of points in the given anomalous cluster. In that way some neighbours of the anomalous points will not be in the same cluster and hence the LOF scores of the anomalous points will be large.

Furthermore, variations of the LOF method proposed in this section has been derived, like the one that picks the first n points with the highest outlier scores and sets them as outliers (Aggarwal, 2015). LOF can also be combined with clustering techniques, like the method CBLOF (Cluster-Based Local Outlier Factor) (Aggarwal, 2015).

2.2.4 Advantages and disadvantages

LOF is able to detect local anomalies as the algorithm is based on the local density of the points (Goldstein and Uchida, 2016), by giving each point a degree of outlierness instead

of a binary label which is beneficial for many applications (Breunig et al., 2000; Kriegel et al., 2009; Lee et al., 2011). Nevertheless, it can also find global anomalies, since they also have a low local reachibility density with respect to its neighbours (Goldstein and Uchida, 2016; Lee et al., 2011). Lazarevic et al. (2003) showed that LOF performs better than Mahalanobis distance and other methods in detecting outliers. And Breunig et al. (2000) even states that LOF detects outliers that cannot be identified by using other methods (Breunig et al., 2000).

LOF is very competent in detecting local outliers, however in a dataset that predominately has global outliers, LOF tends to detect too many false positives (normal points detected as anomalies) (Aggarwal, 2015; Goldstein and Uchida, 2016), giving too many false alarms (Goldstein and Uchida, 2016). Hence it should not be used in this type of cases (Goldstein and Uchida, 2016). However if the task is to detect local anomalies, LOF is one of the best candidates for this type of problem (Goldstein and Uchida, 2016).

The LOF method does not make any assumption on the distribution of the data (Kriegel et al., 2009), hence it is more robust than other outlier detection methods that assume specific distributions (Lee et al., 2011).

Furthermore, it is easy to implement and interpret (Aggarwal, 2015). The LOF method is able to detect anomalies in data sets with clusters of different densities as it considers the local densities around points (Aggarwal, 2015; Lazarevic et al., 2003). However if different density regions in the space are not clearly divided it can be impractical (Aggarwal, 2015), as the LOF method gives an erroneous outlier score to the border points of the clusters (Goldstein and Uchida, 2016).

As stated above, changes in the value of $MinPts$ can be very influential in the performance of the algorithm as it is very sensitive to the choice of this parameter (Kriegel et al., 2009). Hence it is not as robust as other methods (Aggarwal, 2015). An erroneous choice in the parameter $MinPts$ can lead to unstable results (Kriegel et al., 2009). This is a clear issue as for unsupervised methods is more important to be robust across different parameter values and data sets as the ground truth is unknown (Aggarwal, 2015). However at the best possible value of $MinPts$, better results can be obtained than other anomaly detection algorithms (Aggarwal, 2015). LOF outperforms other density-based outlier methods across multiple data sets when the appropriate value of $MinPts$ is chosen (Kriegel et al., 2009).

As LOF gives a score of outlierness to each point, it is more flexible than other binary label methods for detecting outliers (Kriegel et al., 2009). Being an outlier is not a binary characteristic (Breunig et al., 1999), but rather it is a property that it is represented by a scoring system which corresponds to the degree of how outlying is a point with respect to its nearest clustering structure (Breunig et al., 1999). This local notion is more suitable for determining different types of outliers (Breunig et al., 1999). And real world applications data sets often have more complex structures and observations are only outliers with respect to its local neighbours (Breunig et al., 1999).

However the scales of the LOF values can vary from one data set to another, and a point which is an outlier in a data set it can be normal in the other with the same outlying score (Kriegel et al., 2009), which makes it hard to set the threshold score in absence of the ground truth. Also the interpretation of this scores can be difficult (Kriegel et al., 2009). This problem can be solved by using the LoOP algorithm which is a variation

of the LOF method, where each outlying score is set to be between [0,1], and hence it is easily interpretable (Kriegel et al., 2009). The method is able to detect local outliers by giving each point a degree of how outlying it is with respect to its neighbours (Lee et al., 2011). It is not clear which threshold has to be given to distinguish a normal point from an anomaly (Goldstein and Uchida, 2016), as discussed above.

In LOF, the Euclidean distance is used for computing the $MinPts$-nearest neighbours, this assumes that around the point the distribution of the data is spherical (Goldstein and Uchida, 2016). If this assumption is not satisfied, then the density estimation is erroneous (Goldstein and Uchida, 2016). Nevertheless, the method of LOF works for any given distance metric, so it can be chosen appropriately for each application (Duan et al., 2007).

One of the problems with this method is that it is computationally expensive as it requires $O(n^2)$ computations in the worst case (Aggarwal, 2015). Finding the nearest neighbours takes a complexity of $O(n^2)$, the rest of computations for LOF are ignored as they take less than 1% in runtime (Goldstein and Uchida, 2016).

2.3 Density-based spatial clustering of applications with noise (DBSCAN)

2.3.1 Introduction and basic definitions

The Density-Based Spatial Clustering and Application with noise (DBSCAN) method is an unsupervised algorithm proposed by Ester et al. (1996). It was introduced as a density-based technique to discover clusters of arbitrary shape and large size in spatial data sets containing outliers (Çelik et al., 2011; Han et al., 2011; Khan et al., 2014; Mehrotra et al., 2017; Sawant, 2014; Schubert et al., 2017; Sander et al., 1998). It was a promising algorithm as in 2014 it received the SIGKDD test-of-time award (Gan and Tao, 2017; Schubert et al., 2017). It was an answer to the following problems (Ester et al., 1996; Sander et al., 1998):

- The need of domain knowledge to determine the parameters as most of the time for large data sets these values are unknown.
- Not being able to detect clusters of different shapes and sizes in other more traditional clustering algorithms.
- The low efficiency for large data sets of other methods proposed at the time.

DBSCAN uses the conventional density definition, which is the number of observations per unit of volume (Mehrotra et al., 2017). It has two input parameters: $MinPts \in \mathbb{N}$ and *epsilon*, $\epsilon \in \mathbb{N}$ that need to be determined by the user (Sawant, 2014). The value of $MinPts$ represents a threshold for the number of neighbours of a point within an ϵ radius (Schubert et al., 2017).

There are some definitions and some concepts that need to be introduced before developing the algorithm. The first definition is the ϵ-neighbourhood of a point which is similar to the $MinPts$-distance neighbourhood of a point encountered when defining

the LOF method, but replacing the $MinPts$-distance with the parameter ϵ. It is defined as follows (Ester et al., 1996; Han et al., 2011; Khan et al., 2014; Sander et al., 1998):

Definition 2.3.1. Let X be the data set. Then the ϵ-neighbourhood of a point $p \in X$ is defined as follows:

$$N(p) = \{q \in X : d(p, q) \leq \epsilon\}$$

The density of a point is defined as the number of observations in the ϵ-neighbourhood of the point (Sawant, 2014). The main idea is that each point in a cluster, its neghbourhood of radius ϵ has to have at least $MinPts$ number of observations (Khan et al., 2014). Hence, the neighbours of a point are the observations at an ϵ distance from the point (Çelik et al., 2011). Based on the ϵ-neighbourhood the algorithm distinguishes between three types of points (Çelik et al., 2011; Ester et al., 1996):

- **Core point:** It is a point $p \in X$ that satisfies that $|N(p)| \geq MinPts$.
- **Border point:** It is a point $q \in X$ that satisfies $|N(q)| < MinPts$ and $q \in N(p)$ for some core point p.
- **Noise point:** It is a point $q \in X$ that satisfies $|N(q)| < MinPts$ and $q \notin N(p)$ for every core point p.

The algorithm classifies the points as core points, border points or outlier points (Çelik et al., 2011). Core points are those that have at least $MinPts$ number of points in its ϵ-neighbourhood (Çelik et al., 2011; Han et al., 2011; Khan et al., 2014; Mehrotra et al., 2017; Schubert et al., 2017), hence they exceed a density threshold (Campello et al., 2020; Sander et al., 1998). Core points constitute the central part of the clusters (Mehrotra et al., 2017). The algorithm does not limit the number of observations a core point should have in its ϵ-neighbourhood, as a result, this allows clusters that are formed to have arbitrary shapes and size (Sawant, 2014). Border points are the ones that are in the ϵ-neighbourhoods of core points but do not satisfy the core point condition (Çelik et al., 2011). Hence, the ϵ-neighbourhood of a border point will have significantly less points than the one of a core point (Ester et al., 1996). The border points are the rest of the points in the cluster that are not core points (Campello et al., 2020; Mehrotra et al., 2017). Finally, the rest of the points that are neither core points nor border points are labelled as noise (Çelik et al., 2011). These noise points do not belong to any cluster (Campello et al., 2020; Mehrotra et al., 2017).

Next, the concepts of direct density reachability and density reachability are introduced to be able to form clusters. Direct density reachability is defined as follows (Ester et al., 1996; Han et al., 2011; Sander et al., 1998):

Definition 2.3.2. Let X be the data set. A point $q \in X$ is directly density reachable from $p \in X$ if:

(i) $q \in N(p)$

(ii) $\mid N(p) \mid \geq MinPts$

Note that the second condition for direct density reachability is just the condition of being a core point for the point p (Ester et al., 1996). So every neighbour of a core point p is directly reachable from p. This condition is symmetric for core points (Ester et al., 1996) as both satisfy the condition in Definition 2.2.2, hence both are directly reachable from each other. However the condition is asymmetric for a border point and a core point (Ester et al., 1996) as the border point never satisfies the second condition of Definition 2.2.2, due to its border point nature, therefore only the border point is directly density reachable from the core point. Note that noise points are never directly density reachable from any point. This definition of direct density reachability can be extended to density reachability as follows (Ester et al., 1996; Sander et al., 1998):

Definition 2.3.3. A point q is density reachable from p if there exists a path of points $r_1, r_2, ..., r_n$ with $r_1 = p$ and $r_n = q$ such that each r_{i+1} is directly density reachable from r_i.

This extension is still not symmetric for every point (Ester et al., 1996). For two core points it is clear that the relation is symmetric (Ester et al., 1996) as for the Definition 2.2.3 to hold every point except from r_n must be a core point, but $r_n = q$ is now a core point, and as we saw above directly-density reachable condition is symmetric for core points, and hence the density-reachable condition will also be symmetric for core points. On the other hand if $r_n = q$ is a border point then the relationship is not symmetric as r_{n-1} is not directly density reachable from a border point q by Definition 2.2.2 above. With the Definition 2.2.2 and 2.2.3, the relationship between two core points and between a core point and a border point is covered. However, none of the definitions stated above explain the relation between two border points, as they are not directly density reachable or density reachable from each other (Ester et al., 1996), hence to be able to form clusters, the concept of density connectedness is introduced (Campello et al., 2020; Ester et al., 1996; Han et al., 2011; Sander et al., 1998):

Definition 2.3.4. Let X be the data set. A point q is density connected to a point p if there exists a point o such that both q and p are density reachable from o.

This has the advantage of being a symmetric relation for every point that is not a noise point (Ester et al., 1996). And will conform an important part when defining the concept of cluster in the framework of the DBSCAN algorithm. Now the concepts of cluster and noise can be defined more formally in the framework of DBSCAN with these concepts defined above (Ester et al., 1996; Sander et al., 1998):

Definition 2.3.5. Let X be the data set. A cluster C is a non-empty subset of X that satisfies the following two properties:

(i) **Maximality:** $\forall q, p \in X$, if $q \in C$ and p is density reachable from q then $p \in C$.

(ii) **Connectivity:** $\forall q, p \in C$, p and q are density connected.

Definition 2.3.6. Let X be the data set. The noise is a subset, N of X. Having clusters $C_1, C_2, ..., C_n \subset X$, the noise subset is defined as follows:

$$N = \{q \in X | \forall j \in \{1, 2, .., n\} : q \notin C_j\}$$

2.3.2 Tuning the parameters $MinPts$ and ϵ

The Euclidean distance metric is the most common distance metric to be chosen for DBSCAN (Gan and Tao, 2017), however other distance metrics can also be used depending on the application. Schubert et al. (2017) states that DBSCAN was not designed just to employ as distance metric the Euclidean distance but also other metrics. Aside from the distance metric, DBSCAN has two parameters that need to be determined, the parameter $MinPts$ which is the minimum number of points and ϵ which is the neighbourhood distance (Çelik et al., 2011). The ϵ is a positive real number and $MinPts$ is a small positive integer (Gan and Tao, 2017). The two parameters are specified by the user before executing the algorithm (Çelik et al., 2011; Khan et al., 2014). Together, the parameters $MinPts$ and ϵ specify a density level (Campello et al., 2020). The ϵ is the distance threshold and $MinPts$, the density threshold (Campello et al., 2020).

Çelik et al. (2011), who assesed the effects of changes in the parameters *epsilon* and $MinPts$ concluded that as the *epsilon* value increases the anomalies detected by the algorithm decreases and as $MinPts$ increases the number of anomalies found increases. As clusters are formed if there are $MinPts$ number of points in an *epsilon* neighbourhood, the greater $MinPts$ the smaller the chances the *epsilon* neighbourhood will contain these number of points.

The parameters of $MinPts$ and ϵ depend on the aim of the application and on the data set (Mehrotra et al., 2017; Sander et al., 1998). However, it is often complicated to choose a value for these parameters (Mehrotra et al., 2017; Ester et al., 1996; Sander et al., 1998). These parameters are chosen globally (Ester et al., 1996; Sander et al., 1998). Hence, Ester et al. (1996); Sander et al. (1998) propose to set these values to be the parameters of the normal cluster with less density. If these parameters are found then the algorithm will be able to find the other clusters in theory. This is because if the epsilon of the less dense cluster is ϵ', then $\epsilon' \geq \epsilon_j$ for $j \in \{1, ..., K\}$ where K is the total number of clusters of the database and ϵ_j represents the epsilon of each cluster. At the same time if $MinPts'$ is the $MinPts$ of the less dense cluster, then $MinPts \leq_j$ for $j \in \{1, ..., K\}$ where $_j$ represents the parameter $MinPts$ of each cluster. The method is based on the k-distance graph which is defined in the following section.

The sorted k-distance graph

Ester et al. (1996); Sander et al. (1998); Sawant (2014) define the k-distance for $k \geq 1$ as a function that maps each point to the distance to its k^{th} nearest neighbour. Then the sorted k-distance graph is the the k-distance of every point in the data set sorted in descending order (Sander et al., 1998). Now by setting ϵ to the k-distance(p) representing the k-distance of a point $p \in D$ and $MinPts = k$, every point with a k-distance less than or equal to k-distance(p) will be detected as core points as they will contain in their ϵ-neighbourhood at least k points (Ester et al., 1996; Sander et al., 1998). Then if the point belonging to the less dense cluster could be found, the parameter values for the DBSCAN algorithm would be determined (Sander et al., 1998).

Before being able to find this object the value of k and hence the value of $MinPts$

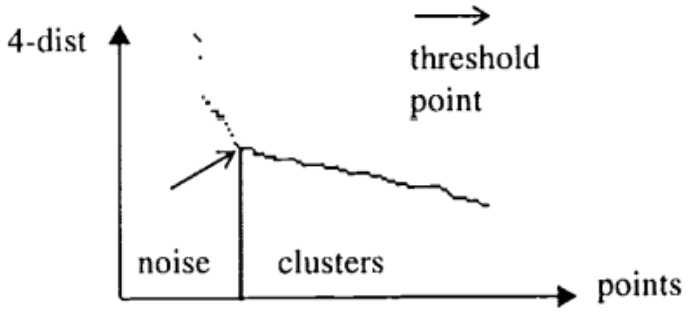

Figure 2.2: Plot of the sorted k-distance graph, $k = 4$ (Ester et al., 1996)

needs to be determined (Sander et al., 1998). Smaller values of k will be beneficial as the computational cost will be lower when computing the k-distance values for each point in the data set (Sander et al., 1998). Ester et al. (1996) propose $MinPts = 4$ for bivariate data as they did not find many differences in their results when setting $MinPts = 4$ than when setting $MinPts > 4$. On the other hand for multivariate data (Sander et al., 1998) propose $MinPts = 2 \cdot q$, where q is the dimension of the data set. If the data contains anomalous clusters it is preferable to set a large value of $MinPts$ in order to identify these anomalies.

Once the $MinPts$ parameter is determined, an appropriate value for ϵ needs to be given. If a point belonging to the less dense cluster with a high k-distance is found, then the parameter ϵ can be set to be this value (Ester et al., 1996; Sander et al., 1998). The experiments of Sander et al. (1998) illustrate that this point is close to the first "valley" of the sorted k-distance plot. Every point to the left of this threshold will have a higher k-distance value and will be assigned to noise (Sander et al., 1998; Sawant, 2014). On the other hand the points to the right of this threshold will have a lower k-distance value and will be a part of some cluster (Sander et al., 1998; Sawant, 2014). This valley is easily identified when represented graphically (Ester et al., 1996; Sander et al., 1998; Sawant, 2014).

The robustness of the method for estimating the parameter ϵ is the width of the range of correct ϵ values that can be used in the algorithm and this robustness highly relies on the application (Sander et al., 1998). Nevertheless, this range is broad enough to be able to determine the ϵ value from a small sample containing between 1% to 10% of the points of the whole database (Sander et al., 1998).

The shape of the sorted k-distance plot depends on the distribution of the k-nearest neighbours distances of the points (Sander et al., 1998; Sawant, 2014). Being more "stairs-like" if the points are in clusters of varying densities or the first "valley" being less apparent if the density of the clusters and the noise points are similar (Sawant, 2014).

2.3.3 DBSCAN method for detecting anomalies

This algorithm can be applied in anomaly detection (Çelik et al., 2011). Clusters are formed based on the density of neighbourhood point (Han et al., 2011). The clusters are computed iteratively (Campello et al., 2020). DBSCAN picks a random point p (Han

et al., 2011; Sawant, 2014). A cluster is created if the ϵ-neighbourhood of the point p exceeds the $MinPts$ number of observations (Çelik et al., 2011; Khan et al., 2014). So if p is a core point (Campello et al., 2020; Sawant, 2014). If p is not a core point then DBSCAN labels p as noise and chooses another unseen point (Han et al., 2011; Sawant, 2014). Every point connected to the core point p is assigned to the cluster (Campello et al., 2020; Schubert et al., 2017). Or stated in another way points that are density-reachable from observations already in the cluster are added until there is no new point that can be added to the cluster (Han et al., 2011; Khan et al., 2014; Sawant, 2014; Schubert et al., 2017). This constitutes a cluster and the whole algorithm stops when no new point can be added to any cluster (Han et al., 2011; Khan et al., 2014). The algorithm identifies each point as either a noise point or as part of a cluster (Sawant, 2014). The DBSCAN locates core objects, which are points with dense neighbourhoods (Gan and Tao, 2017; Han et al., 2011). These areas are separated from regions of lower density (Schubert et al., 2017). The noise points returned by the algorithm will be the anomalies and the observations belonging to clusters the normal points. More formally the method is described as follows (Çelik et al., 2011; Ester et al., 1996; Han et al., 2011; Khan et al., 2014; Mehrotra et al., 2017; Schubert et al., 2017):

1. Set the distance metric and the parameters $MinPts$ and ϵ following the established criterion in Section 2.
2. Select a random unseen point $p \in D$ and find the set $N_\epsilon(p)$.
   (a) If $|N_\epsilon(p)| < MinPts$, label p as noise and as seen. Go back to step 2.
   (b) Else, begin the formation of the cluster C.
      i. Add the point p to C.
      ii. Add the points $q \in N_\epsilon(p)$ to the set C iff they have not been assigned to another cluster.
      iii. For each $q \in N_\epsilon(p)$ satisfying $|N_\epsilon(q)| \geq MinPts$ add all the points $r \in N_\epsilon(q)$ to the cluster C iff they have not be assigned to another cluster.
      iv. Repeat this process until $\forall c \notin C$ and c not belonging to another cluster, $c \notin N_\epsilon(i)$ for $i \in C$ and $|N_\epsilon(i)| > MinPts$. Then the cluster C is complete.
      v. Label every point in the cluster as seen.
3. Go to step 2 and start again with an unseen point. Stop when every point is labelled as seen.
4. For a point $t \in D$
   (a) If $t \in K$ for a cluster $K \subset D$ formed by the DBSCAN procedure label t as non-anomalous.
   (b) Else, label t as anomalous.

There can be cases where a cluster only consists of a core point and the border points contained in its ϵ-neighbourhood (Campello et al., 2020). By setting a minimum number of points requirement for each cluster, the method is able to detect anomalous clusters that would be missed otherwise (Campello et al., 2020). Hence the method is able to

detect clusters of anomalies if these groups of points do not satisfy the condition of having $MinPts$ number of points (Çelik et al., 2011).

2.3.4 Advantages and disadvantages

DBSCAN unlike other clustering algorithms, it is able not only to detect clusters but also the outliers that do not belong to any of those clusters (Çelik et al., 2011). The algorithm is able to discover anomalies between clusters of normal points, unlike other statistical methods that are only able to detect extreme outliers that are above or below certain threshold (Çelik et al., 2011). Hence, DBSCAN is able to detect both the extreme outliers and the ones between clusters of normal points as outliers are not only extreme values but also are those that do not occur frequently (Çelik et al., 2011). Furthermore, it is able to detect small clusters of anomalies if the number of points in the cluster is less than the parameter value $MinPts$ (Khan et al., 2014).

The algorithm is able to find clusters of different shapes and sizes (Han et al., 2011; Khan et al., 2014; Sander et al., 1998; Sawant, 2014). However it has problems in differentiating clusters that are at a close distance from each other (Khan et al., 2014). Moreover, clustering becomes problematic when the data set has clusters of varying densities as the parameters are chosen globally (Khan et al., 2014; Mehrotra et al., 2017).

One of the drawbacks of DBSCAN is giving appropriate values to the $MinPts$ and ϵ parameters as the ground truth is unknown (Khan et al., 2014). The choice of the input parameters has a high influence on the result of the algorithm, however they are often hard to determine (Han et al., 2011; Sawant, 2014). Furthermore, this parameters must be globally determined and for many data sets, this choice does not describe the clustering structure in an accurate way (Sawant, 2014). For these reason the performance of the algorithm in multi-density data is poor (Sawant, 2014). In this case, non-anomalous clusters may merge together and some clusters may be detected as anomalies (Sawant, 2014). However, the method is able to find clusters of different shapes and sizes if the appropriate parameters of $MinPts$ and ϵ are selected (Han et al., 2011). It has the advantage that unlike other clustering methods, the number of clusters does not need to be known for the evaluation of the algorithm (Ester et al., 1996; Mehrotra et al., 2017; Sawant, 2014). Nevertheless Mehrotra et al. (2017) argues that tuning the parameter for the number of clusters can be more challenging than determining ϵ.

Khan et al. (2014) states the computational complexity of the algorithm, when encountered with data sets of large size, is very costly. However Ester et al. (1996); Sander et al. (1998) consider that DBSCAN is an effective algorithm even if the size of the data set is large. Ester et al. (1996) stated that the average run time complexity was $O(nlog(n))$, however they never gave a proof of such statement. Many authors such as Gan and Tao (2017); Han et al. (2011) have specified that the computational complexity of the algorithm is $O(nlog(n))$ if spatial indexes are used and $O(n^2)$ otherwise.

2.4 Gaussian Mixture Model (GMM)

2.4.1 Introduction

The Gaussian Mixture Model (GMM) is a mixture model where the different components are Gaussian distributions (Bishop, 2006; Murphy, 2012). Each one of these Gaussian distributions will conform a **class** in the GMM with mean μ_k and co-variance matrix Σ_k for $k \in \{1, ..., K\}$, where K is the total number of classes (Murphy, 2012).

For each class $k \in \{1, ..., K\}$ we will have to determine the prior probability that any data point will belong to that class before seeing any observation. The prior probability of a class k for any observation x_i is defined as (Bishop, 2006):

$$\pi_k = p(z_i = k) \tag{2.10}$$

where z_i is a latent variable indicating to which class each observation belongs (Murphy, 2012).

These prior probabilities $\pi_1, \pi_2, ..., \pi_K$ are called the **mixing weights** (Murphy, 2012) of the Gaussian Mixture Model. As these are just probabilities they satisfy $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$.

Along with the prior probability, we have the likelihood of each Gaussian distribution, which measures the probability that knowing a specific class k an observation x_i will belong to that class. It is defined as follows (Bishop, 2006):

$$p(x|\mu_k, \Sigma_k) = N_q(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{q}{2}}|\Sigma_k|^{\frac{1}{2}}} \, exp(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)) \tag{2.11}$$

where $N(x|\mu_k, \Sigma_k)$ is just the q-multivariate normal distribution with mean μ_k and co-variance matrix Σ_k.

As the GMM is just a combination of Gaussian distributions, to obtain the likelihood that an observation x_i will be in the Gaussian Mixture Model we just need to sum the likelihood of x_i for all classes and multiply each one of them by its mixing weight. We introduce this mixing weights in the model as in the real world we will not normally have the same proportions of data for each class so they will act as a normalising factor so that the Mixture Model integrates to one.

Then, the Gaussian Mixture Model of K classes has the form (Murphy, 2012):

$$p(x|\pi, \mu, \Sigma) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \tag{2.12}$$

The model is now defined as long as the parameters π_k, μ_k and Σ_k are known for all the classes. The problem in real world applications knowing these parameters is not normally the case, so they need to be estimated. Hence the EM-algorithm is introduced.

2.4.2 The EM algorithm

The EM algorithm was proposed by Dempster et al. (1977). If x_i is an observation and z_i is the latent variable, then to estimate the parameters μ, Σ and π the log likelihood needs to be maximised (Murphy, 2012). For simplicity when deriving these algorithm the parameters will be encompassed in the parameter θ, so $\theta = (\pi_k, \mu_k, \Sigma_k)$. The log likelihood function is defined as follows (Murphy, 2012; Nguyen, 2015):

$$\ell(\theta) = \sum_{i=1}^{n} \log p(x_i|\theta) = \sum_{i=1}^{n} \log \left[\sum_{z_i} p(x_i, z_i|\theta) \right] \tag{2.13}$$

However this is complicated to maximise as the log cannot be moved inside the sum (Murphy, 2012). Hence, to avoid this problem the EM algorithm defines the complete data log likelihood (Murphy, 2012; Nguyen, 2015). It is as follows (Murphy, 2012; Nguyen, 2015):

$$\ell_c(\theta) \triangleq \sum_{i=1}^{n} p(x_i, z_i|\theta) \tag{2.14}$$

This likelihood cannot be calculated as the z_i is unknown. For this reason Murphy (2012); Nguyen (2015) defines the expected complete data log likelihood as follows:

$$Q(\theta, \theta^{t-1}) = \mathbb{E}[\ell_c(\theta)|X, \theta^{t-1}] \tag{2.15}$$

Here X is the data matrix, t is the iteration number and Q is the auxiliary function (Murphy, 2012). The main goal of the E-step is to determine $Q(\theta, \theta^{t-1})$ (Murphy, 2012; Nguyen, 2015). On the other hand, the M-step optimizes Q with respect to θ as follows (Murphy, 2012; Nguyen, 2015):

$$\theta^t = \arg\max_{\theta} Q(\theta, \theta^{t-1}) \tag{2.16}$$

Once these definitions are stated, the expected complete data log likelihood can be derived as follows (Murphy, 2012):

$$Q(\theta, \theta^{t-1}) \triangleq \mathbb{E}\left[\sum_{i=1}^{n} \log p(x_i, z_i|\theta) \right] = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \log(\pi_k) + \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \log N(x_i|\theta_k) \tag{2.17}$$

In the E-step, the responsibilities r_{ik} are computed (Murphy, 2012):

$$r_{ik} = \frac{\pi_k N(x_i|\theta_k^{(t-1)})}{\sum_{j=1}^{K} \pi_j N(x_i|\theta_j^{(t-1)})} \tag{2.18}$$

In the M-step as explained above the Q function is optimized with respect to the parameters π_k, μ_k, Σ_k. As a result the following parameter estimates are obtained (Murphy, 2012):

$$\mu_k = \frac{1}{\sum_{i=1}^{n} r_{ik}} \sum_{i=1}^{n} r_{ik} x_i \tag{2.19}$$

$$\Sigma_k = \frac{1}{\sum_{i=1}^{n} r_{ik}} \sum_{i=1}^{n} r_{ik} (x_i - \mu_k)(x_i - \mu_k)^T \tag{2.20}$$

$$\pi_k = \frac{\sum_{i=1}^{n} r_{ik}}{n} \tag{2.21}$$

After estimating the parameters π_k, μ_k, Σ_k for each $k \in \{1, ..., K\}$ with the equations defined above, we set $\theta^t = (\pi_k, \mu_k, \Sigma_k)$ and go to the E-step and start the algorithm with these new parameters (Murphy, 2012). The method can be summarised as follows (Bishop, 2006):

1. Initialize with some random values for the parameters μ_k, Σ_k, π_k for each class.
2. **Expectation Step (E)**:Compute the **responsibilities**, r_{ik}, for $k \in \{1, .., K\}$ and $i \in \{1, ..., n\}$, being n the total number of observations.

$$r_{ik} = \frac{\pi_k N(x_i|\mu_k \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i|\mu_j, \Sigma_j)}$$

3. **Maximization Step (M)**: Recompute the parameters π_k, μ_k and Σ_k for all $k \in \{1, .., K\}$. We will obtain:

$$\mu_k = \frac{1}{\sum_{i=1}^{n} r_{ik}} \sum_{i=1}^{n} r_{ik} x_i$$

$$\Sigma_k = \frac{1}{\sum_{i=1}^{n} r_{ik}} \sum_{i=1}^{n} r_{ik} (x_i - \mu_k)(x_i - \mu_k)^T$$

$$\pi_k = \frac{\sum_{i=1}^{n} r_{ik}}{n}$$

4. Repeat step 2 and step 3 until given an ϵ value:
   $|\log\left[p(x_i, \theta^{(t+1)}\right] - \log\left[p(x_i, \theta^{(t+1)}\right]| < \epsilon$

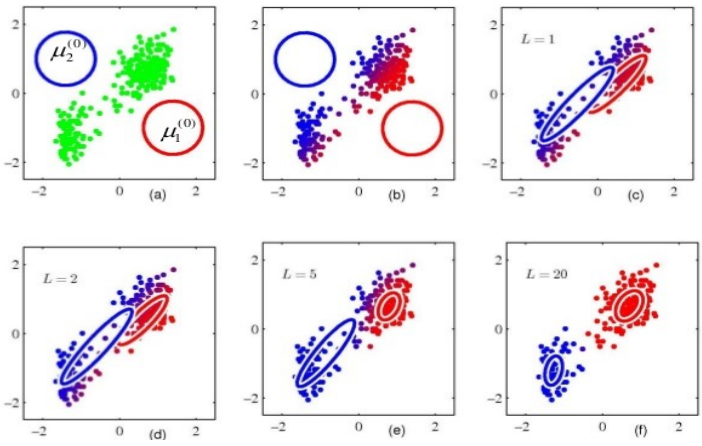

Figure 2.3: Example of EM algorithm with K=2, where L are the number of iterations (Murphy, 2012)

2.4.3 GMM methods for anomaly detection

After defining the model and estimating the unknown parameters we can apply the Gaussian Mixture Model to anomaly detection. There are two approaches we can take to detect possible outliers: density estimation and clustering (Aggarwal, 2015).

Anomaly detection using GMM Clustering

In **GMM Clustering**, each Gaussian distribution will be a class or cluster. In general, for anomaly detection there will be two classes, one grouping the anomalies and the other the normal points.

Then the Gaussian Mixture Model can be simplified as follows:

$$p(x|\pi,\mu,\Sigma) = \pi_1 N(x|\mu_1,\Sigma_1) + (1-\pi_1)N(x|\mu_2,\Sigma_2)$$

For assigning each point to a particular class the posterior probability that point x_i belongs to a specific cluster k must be computed. This is called the **responsibility**, r_{ik}, that cluster k takes for data point x_i and is defined as follows (Murphy, 2012):

$$r_{ik} = p(Z_i = k|x_i,\pi,\mu,\Sigma) = \frac{\pi_k \;\; N(x_i|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \;\; N(x_i|\mu_j,\Sigma_j)}$$

As GMM performs soft clustering, which means that each point has a probability for belonging to each one of the clusters, each point x_i will be assigned to the class that has more responsibility, r_{ik}, for that point x_i (Murphy, 2012).

Furthermore, GMM is a density based model so it has the advantage that can detect clusters of different sizes and shapes (Bishop, 2006; Murphy, 2012).

Although apparently in the detection of anomalies it seems that there are only two clusters, one for the normal points, and the other for the anomalous points, the issue presents some controversies.

In the first place, to carry out this subdivision in a simple way, it should be true that at least some anomalous points share common characteristics among themselves, so that a possible classification of the model into two subsets is easy: one cluster having the normal points and the other the anomalous points (with common characteristics).

However this is not always the case since we can find models in which the various anomalies are widely dispersed among themselves and will not be part of the same cluster of anomalies. This may mean that in the sub-classification there can be a cluster of normal points containing anomalous points, as well as a cluster of anomalous points containing normal points. Nevertheless, reaching to the conclusion that we have several normal clusters or several anomalies clusters is not always easy, specially if we cannot visualize the data which will be the case in general as the real world data tends to be multivariate, hence we will set the algorithm to run for two clusters as we want to classify the points into normal and anomaly and see the performance of the algorithm. We will discuss further details in the simulation chapter where we will explore a normal cluster and two smaller cluster of anomalies.

Anomaly detection using GMM Density Threshold

In our model we have defined the likelihood that a specific data point will belong to the Gaussian Mixture Model. Given that anomalies are data points that do not share the same distribution as the rest of the points, their density probabilities under the model will be very low. In this way we can establish a threshold probability in which we distinguish between an anomalous point and a normal one. Instead of the $p(x|\pi, \mu, \Sigma)$ of the model, we will use the logarithm of this probability to decrease the relative variance between normal points to avoid missclassifying normal points as outliers (Aggarwal, 2015).

The method described by Reddy et al. (2017) is as follows:

1. For each point x in the dataset calculate the probability density function(PDF) of the GMM
2. The outlier score could be established directly from the PDF but the probabilities between points with low density and outliers or anomalies will be very similar. For that reason to make this differences greater, define the **outlier score**, OS_x, as follows:
$$OS_x = (log(p(x|\pi, \mu, \Sigma))^2$$
3. Scale the outlier score between 0 and 10 to make it easier for interpretation:
$$ScOS_x = \frac{OS_x - min(OS_x)}{max(OS_x) - min(OS_x)} \cdot 10$$
4. Detect point x as an anomaly if $ScOS_x \geq 8$

This method establishes an outlier score for each point and they establish that points with an outlier score of 9 or 10 can be considered very anomalous. While points that have an score from 0 to 1 indicates a normal point. They set the threshold to be 8 but they do not specify if there can be anomalies with scores between 1 and 8. This

method will have problems when the maximum outlier score $max(OS_x)$ is much greater in scale than the other outlier scores in the dataset, as the denominator of the scale outlier score $ScOS_x$ will be much bigger than the numerator times 10 for most of the points and hence it will lead to most of the points having a $0 \leq ScO_x \leq 1$ and this will result in not detecting the corresponding anomalies for that dataset. To correct this problem we can multiply by 100 instead of 10 when the $max(OS_x)$ highly differs in scale with respect to the rest of Outlier scores. Note that the $max(OS_x)$ will be detected as an anomaly. There can be also be derivations of this method as if our main focus is the detection of anomalies, we can change our previous threshold from 8 to 1. As normal points will tend to have scores between 0 and 1, so we can consider anything that surpasses that threshold to be anomalous.

Other authors, like Emmott et al. (2015) use $-log(p(x))$ as a score, being $p(x)$ the PDF of the point x for the mixture model. Although they give an ensemble approach by training some data, hence it is supervised. For these reason the method that will be applied will be method described by Reddy et al. (2017), however taking the consideration explained above of multiplying the score by 100 instead of by 10.

2.4.4 Different covariance parametrisations of GMM

For the estimation of the covariance matrix in the EM algorithm there are different models that can be used. The package Mclust in R performs GMM Clustering on the data for every possible model for the estimation of the covariance matrix and chooses the more optimal one based on the BIC score which accounts for the model complexity and goodness of fit (Fraley et al., 2012). Simpler models can be used for estimating the covariance matrices of each class by taking into account the geometric features of the clusters (Erar, 2011). These models are derived based on the eigenvalue decomposition of the covariance matrix (Erar, 2011). The eigenvalue decomposition for the covariance matrix of class k is defined as follows (Erar, 2011):

$$\Sigma_k = \lambda_k D_k A_k D_k^T \tag{2.22}$$

Here A_k is the diagonal matrix with the normalized eigenvalues and $|A_k| = 1$. D_k is an orthogonal matrix of eigenvectors and λ_k is a scalar (Erar, 2011).

The covariance models can be classified into three main groups: spherical models, diagonal models and general models (Erar, 2011). In total there are 14 models, out of these only 9 are in closed form and are the ones that will be defined (Erar, 2011). By implementing some of these models the number of unknown parameters decreases significantly, which makes the complexity of the EM algorithm and hence the clustering method much lower (Erar, 2011). The parameter $\alpha = Kq + K - 1$ will be the number of parameters in the means, where K is the total number of clusters and $\beta = \frac{q(q+1)}{2}$ the number of parameters of the full covariance matrix (Erar, 2011). In the case of anomaly detection with $K = 2$, $\alpha = 2q + 1$.

Spherical Family

These models assumes that the variables of every class have the same variance (Erar, 2011). The covariance matrix of each class is a diagonal matrix with the same entry in

the diagonal (Erar, 2011). This results in the covariances being sperical (Erar, 2011). In these models the scale can vary, however they are rotationally invariant. They can be defined as follows (Erar, 2011):

1. **Model EII.** In this model every class has the same volume and shape. The covariance matrix for every class is $\Sigma = \lambda I$, where $I \in \mathbb{R}^{||\times||}$. The total number of parameters that need to be estimated are $\alpha + 1 = 2q + 2$ for $K = 2$.
2. **Model VII.** The covariance matrix for class k is $\Sigma_k = \lambda_k I$. Hence each class has a different value of λ, however their covariance matrix is still diagonal with the same entries on the diagonal. The total number of parameters that need to be determined is $\alpha + q = 3q + 1$ for $K = 2$.

Diagonal Family

These Diagonal models constrain $B = DAD^T$, here B is the diagonal matrix that satisfies $|B| = 1$ (Erar, 2011). The shape and the volume of the covariance matrix of each class is determined by λ and B (Erar, 2011). The Diagonal models are defined as follows (Erar, 2011):

1. **Model EEI.** The covariance matrix for this model is $\Sigma = \lambda B$. In this model the clusters will have fixed volume and shape. The total number of parameters that need to be determined is $\alpha + q = \alpha + 2 = 2q + 3$ for $K = 2$.
2. **Model EVI.** This model sets the covariance metric for a class k to be $\Sigma_k = \lambda B_k$. Each class has a different diagonal matrix B, while λ remains constant for every class. The total number of parameters that need to be estimated is $\alpha + Kq - K + 1 = \alpha + 2q - q + 1 = 3q + 2$ for $K = 2$.
3. **Model VVI.** The covariance matrix for class k is $\Sigma_k = \lambda_k B_k$. Each class can have a different diagonal matrix B and a different parameter λ(YO). This implies that the orientation of the clusters is fixed but there can be variations on the volume and shape of these clusters. The total number of parameters that need to be determined is $\alpha + Kq = \alpha + 2q = 4q + 1$ for $K = 2$.

General Family

These models are the most general and they are not contrained to have diagonal covariance matrices (Erar, 2011). They are defined as follows(Erar, 2011):

1. **Model EEE.** The assumed covariance matrix for every class in this model is $\Sigma = \lambda DAD^T$. Hence the shape,volume and orientation is fixed for every cluster under this model. This implies that the non-diagonal entries of the covariance matrices have to be non-zero. The total number of parameters that need to be estimated is $\alpha + \beta = 2q + 1 + \frac{q(q+1)}{2}$ for $K = 2$.
2. **Model EEV.** The covariance matrix for class k is $\Sigma_k = \lambda D_k A D_k^T$. Hence, the shape and volume of the clusters is fixed, but the orientation of the cluster can vary. The total number of parameters that need to be determined is $\alpha + K\beta - (K-1)q = q^2 + 2q + 1$ for $K = 2$.

3. **Model EVV.** The covariance matrix is $\Sigma_k = \lambda D_k A_k D_k^T$ for class k. Here, the volume of the clusters is fixed while their orientation and shape can vary. The total number of parameters that need to be determined is $\alpha + K\beta - (K-1) = q^2 + 3q$ for $K = 2$.

4. **Model VVV.** The covariance matrix for class k is $\Sigma_k = \lambda_k D_k A_k D_k^T$. It is the unconstrained model where the total number of parameters that need to be estimated is $\alpha + K\beta = q^2 + 3q + 1$ for $K = 2$.

Chapter 3

Metrics

In this chapter, several metrics are going to be explained to be able to evaluate the methods later on. There are three types of evaluation metrics, the threshold metrics, the probability metrics and the ranking metrics (Hossin and Sulaiman, 2015). The main focus of this project will be in the threshold metrics that can be derived from the *Confusion matrix*.

3.1 Confusion matrix

The confusion matrix summarizes the performance of the algorithm in a binary classification problem (Hossin and Sulaiman, 2015; Kohl, 2012; Luque et al., 2019; Raschka, 2014; Tharwat, 2020). Let P be the label of the *Positive* class and N the label of the *Negative* class (Canbek et al., 2017; Luque et al., 2019; Powers, 2020; Raschka, 2014; Tharwat, 2020). The *Confusion matrix* is represented by the following table (Canbek et al., 2017; Hossin and Sulaiman, 2015; Kohl, 2012; Luque et al., 2019; Parikh et al., 2008; Powers, 2020; Raschka, 2014; Tharwat, 2020):

		Actual class	
		P	**N**
Prediction outcome	**P**	True Positive (TP)	False Positive (FP)
	N	False Negative (FN)	True Negative (TN)
	total	P=TP+FN	N=FP+TN

The rows of the table represent the predicted outcomes and the columns the actual class (Hossin and Sulaiman, 2015; Powers, 2020; Raschka, 2014). It has four possible outcomes, the first diagonal represents the samples that were correctly classified as

Positives and *Negatives*, conformed by the *True Positives (TP)* and the *True Negatives (TN)*, respectively (Canbek et al., 2017; Hossin and Sulaiman, 2015; Parikh et al., 2008; Powers, 2020; Tharwat, 2020). The other diagonal represents the observations that were misclassified as *Positives* and *Negatives* conformed by the *False Positives (FP)* or *Type I error* and the *False Negatives (FN)* or *Type II error*, respectively (Canbek et al., 2017; Hossin and Sulaiman, 2015; Parikh et al., 2008; Powers, 2020; Tharwat, 2020). In our case the frauds will be represented by the *Positives* and the normal points will be represented by the *Negatives*. *Type II errors* can be worse than *Type I errors* in fraud detection, as they are frauds that are not detected ((Canbek et al., 2017)).

Furthermore the total number of true *Positives* is the sum of the *True Postitives* and the *False Negatives* (Luque et al., 2019; Powers, 2020; Tharwat, 2020). And the total number of true *Negatives* is the sum of the *False Positives* and *True Negatives* (Luque et al., 2019; Powers, 2020; Tharwat, 2020).

From the *Confusion matrix* many other classification metrics can be computed (Canbek et al., 2017; Hossin and Sulaiman, 2015; Luque et al., 2019; Tharwat, 2020). As anomalies tend to be fewer than non-anomalous points, the data sets will be imbalanced as the observations of one class will outnumber the observations of the other class (Luque et al., 2019; Tharwat, 2020). When these types of cases occur there will be metrics derived from the *Confusion matrix* that will be sensitive to imbalanced data sets and this can be a major problem (Luque et al., 2019; Tharwat, 2020). The class distribution is represented by the proportion between the *Positive* cases and the *Negative* cases $\frac{P}{N}$ which is an interaction between the left and right column (Tharwat, 2020). Hence the metrics using values in both columns of the *Confusion matrix* will be sensitive to imbalance data sets (Tharwat, 2020).

3.1.1 Accuracy and Prediction Error

The *Accuracy (ACC)* and the *Prediction Error* or *Error (ERR)* are common used metrics (Canbek et al., 2017; Hossin and Sulaiman, 2015; Tharwat, 2020). They are easy to compute and understand (Hossin and Sulaiman, 2015). Both the *Accuracy* and the *Error* give information about how many cases were misclassified (Raschka, 2014). The *Accuracy* is the ratio between the total number of correct predictions and the total number of predictions (Canbek et al., 2017; Chicco and Jurman, 2020; Hossin and Sulaiman, 2015; Raschka, 2014; Tharwat, 2020). While the *Error* is ratio between the number of wrong predictions and the total number of predictions (Hossin and Sulaiman, 2015; Raschka, 2014; Tharwat, 2020). The *Accuracy* (Canbek et al., 2017; Chicco and Jurman, 2020; Hossin and Sulaiman, 2015; Kohl, 2012; Luque et al., 2019; Powers, 2020; Raschka, 2014; Tharwat, 2020) and the *Error* (Kohl, 2012; Raschka, 2014; Tharwat, 2020) are defined as follows:

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR \tag{3.1}$$

$$ERR = \frac{FP + FN}{FP + FN + TP + TN} = 1 - ACC \tag{3.2}$$

The *Accuracy* lies within the range $[0, 1]$ (Canbek et al., 2017; Chicco and Jurman, 2020; Luque et al., 2019). A value of 1 for the *Accuracy* corresponds to perfect prediction

and a value of 0 corresponds to worst possible prediction (Chicco and Jurman, 2020). Additionally, these metrics are not robust to imbalance data sets (Chicco and Jurman, 2020; Hossin and Sulaiman, 2015; Luque et al., 2019; Tharwat, 2020). And cannot be regarded as reliable as they provide an optimistic measure for the capability of the method in detecting the majority class (Chicco and Jurman, 2020; Hossin and Sulaiman, 2015). They are complementary metrics (Hossin and Sulaiman, 2015; Powers, 2020; Tharwat, 2020). However, the *Accuracy* does not give any information about how many of the total correctly classified samples were *Positive* and how many were *Negatives* (Tharwat, 2020).

3.1.2 Balanced Accuracy

An alternative to the *Accuracy* is the *Balanced Accuracy (BA)* (Canbek et al., 2017). The *Balanced Accuracy* is the mean of the *Sensitivity* and the *Specificity*. It is defined as follows (Canbek et al., 2017; Tharwat, 2020):

$$BA = \frac{1}{2}(TPR + TNR) \tag{3.3}$$

The *Balance Error Rate (BER)* is the complement of the *Balanced Accuracy* and it is defined as follows (Tharwat, 2020):

$$BER = 1 - BA \tag{3.4}$$

Both metrics are robust to imbalance data sets (Luque et al., 2019; Tharwat, 2020) and both lie within the range $[0, 1]$ (Canbek et al., 2017).

3.1.3 Sensitivity and Specificity

The *True Positive Rate (TPR)*, *Sensitivity (SENS)* or *Recall* represents the ratio of correctly classified *Positive* cases out of the total number of true *Positives* (Hossin and Sulaiman, 2015; Kohl, 2012; Parikh et al., 2008; Powers, 2020; Raschka, 2014; Tharwat, 2020). On the other hand, the *True Negative Rate (TNR)* or *Specificity (SPEC)* is the proportion of correctly classified *Negative* cases out of the total number of true *Negatives* (Hossin and Sulaiman, 2015; Kohl, 2012; Parikh et al., 2008; Powers, 2020; Raschka, 2014; Tharwat, 2020). They are defined as follows (Canbek et al., 2017; Luque et al., 2019; Hossin and Sulaiman, 2015; Kohl, 2012; Parikh et al., 2008; Powers, 2020; Raschka, 2014; Tharwat, 2020):

$$SENS = \frac{TP}{P} = \frac{TP}{FN + TP} \tag{3.5}$$

$$SPEC = \frac{TN}{N} = \frac{TN}{FP + TN} \tag{3.6}$$

They both lie within $[0, 1]$ (Canbek et al., 2017; Luque et al., 2019). These two metrics are suitable for performance evaluation on imbalance data sets as each one just takes into consideration one class (Luque et al., 2019; Hossin and Sulaiman, 2015; Raschka,

2014; Tharwat, 2020). The *Sensitivity* and *Specificity* can be considered as metrics of accuracy for the true *Positives* and the true *Negatives*, respectively (Tharwat, 2020).

The *Sensitivity* and *Specificity* have complementary metrics which are the *False Positive Rate (FPR)* and the *False Negative Rate (FNR)*, respectively (Powers, 2020; Tharwat, 2020). The *False Positive Rate* represents the proportion of incorrectly classified *Negative* cases out of the total number of true *Negatives* (Powers, 2020; Raschka, 2014; Tharwat, 2020). On the other hand, the *False Negative Rate* is the ratio between the incorrectly classified *Positive* cases and the total number of true *Positives* (Powers, 2020; Tharwat, 2020). They are robust to imbalanced data sets (Raschka, 2014; Tharwat, 2020). Furthermore they lie within the range $[0, 1]$ (Canbek et al., 2017). These metrics are as follows (Canbek et al., 2017; Powers, 2020; Tharwat, 2020):

$$FPR = 1 - SPEC = \frac{FP}{N} = \frac{FP}{FP + TN} \tag{3.7}$$

$$FNR = 1 - SENS = \frac{FN}{P} = \frac{FN}{FN + TP} \tag{3.8}$$

3.1.4 Predictive values

The *Positive Predictive Value (PPV)* or *Precision* is the probability that a case classified as *Positive* is a true *Positive* case (Kohl, 2012; Parikh et al., 2008). And it is the proportion of correctly detected *Positives* out of the total number of *Positives* detected (Hossin and Sulaiman, 2015; Parikh et al., 2008; Powers, 2020; Tharwat, 2020). On the other hand, the *Negative Predictive Value (NPV)* is the probability that a case classified as *Negative* is a true *Negative* (Kohl, 2012; Parikh et al., 2008). Hence, it is the proportion of correctly detected *Negatives* out of the total number of *Negatives* detected (Parikh et al., 2008; Powers, 2020; Tharwat, 2020). They are defined as follows (Kohl, 2012; Luque et al., 2019; Parikh et al., 2008; Powers, 2020; Tharwat, 2020):

$$PPV = \frac{TP}{TP + FP} \tag{3.9}$$

$$NPV = \frac{TN}{TN + FN} \tag{3.10}$$

These metrics are sensitive to imbalance data sets (Luque et al., 2019; Tharwat, 2020). They both lie within the range $[0, 1]$ (Luque et al., 2019). If these values are high then the performance of the method is good (Parikh et al., 2008). The *Precision* and the *Negative Predictive Value* consider only the *Positives* and the *Negatives*, respectively (Hossin and Sulaiman, 2015; Powers, 2020). And for determining the performance of a method both classes must be included (Hossin and Sulaiman, 2015), hence both metrics need to be taken into consideration.

3.1.5 The F_1-Score

The F_1-score combines the metric *Precision* and *Recall* (Raschka, 2014). It is the harmonic mean of these two metrics (Canbek et al., 2017; Chicco and Jurman, 2020;

Hossin and Sulaiman, 2015; Im Walde, 2006; Tharwat, 2020). It is defined as follows (Chicco and Jurman, 2020; Hossin and Sulaiman, 2015; Luque et al., 2019; Powers, 2015, 2020; Raschka, 2014; Im Walde, 2006; Tharwat, 2020):

$$F_1 = \frac{2 \cdot PPV \cdot TPR}{PPV + TPR} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.11}$$

Hossin and Sulaiman (2015) even state that it is a better indicator than the *Accuracy* in determining which method is best. It is within the range $[0, 1]$ (Chicco and Jurman, 2020; Luque et al., 2019; Tharwat, 2020). High values of this metric imply a good performance of the method in predicting the true labels of the observations (Tharwat, 2020). It does not take into account the TN (Chicco and Jurman, 2020; Powers, 2015; Tharwat, 2020). Hence it is not symmetric if the classes are swapped (Chicco and Jurman, 2020) and it only considers one class (Powers, 2015). Furthermore it is sensitive to imbalanced data sets (Luque et al., 2019; Powers, 2015; Tharwat, 2020).

3.1.6 The Geometric Mean

It combines the metrics of *Sensitivity* and *Specificity* in a single metric (Tharwat, 2020). As it only depends on this two metrics it is robust with respect to imbalance data sets (Tharwat, 2020; Luque et al., 2019). It is defined as follows (Tharwat, 2020; Hossin and Sulaiman, 2015; Luque et al., 2019):

$$GM = \sqrt{TPR \cdot TNR} \tag{3.12}$$

The range of GM is $[0, 1]$ (Luque et al., 2019). To obtain a high score in this metric, the values of the *Sensitivity* and the *Specificity* must be maximized, but always maintaining a balance between the two metrics (Hossin and Sulaiman, 2015). Hence a values close to 1 will mean a good performance and values close to 0 a poor performance. It is an alternative metric to the *Accuracy* (Hossin and Sulaiman, 2015; Canbek et al., 2017).

3.1.7 Youden's Index(YI) or Bookmaker Informedness(BM)

It was developed by Youden in 1950 (Sokolova et al., 2006). Also known as *Bookmaker Informedness* (Tharwat, 2020; Luque et al., 2019). It is a common used metric that measures the discriminative power of the method (Tharwat, 2020; Youden, 1950). It assesses the method's capacity to prevent mistakes in labelling the samples (Sokolova et al., 2006). It is a function of the metrics *Sensitivity* and *Specificity* (Tharwat, 2020), and hence it is easily applicable as it does not need further information (Fluss et al., 2005). It links in one metric both the *Sensitivity* and the *Specificity* (Sokolova et al., 2006). It evaluates the performance of the algorithm on the *Positive* and *Negative* samples equally (Sokolova et al., 2006). It is defined as follows (Youden, 1950; Fluss et al., 2005; Sokolova et al., 2006; Tharwat, 2020):

$$YI = TPR + TNR - 1 \tag{3.13}$$

The range of YI is $[0, 1]$, being one if the performance of the method is perfect and 0 if the method performs poorly (Tharwat, 2020; Fluss et al., 2005; Luque et al., 2019).

This metric is robust when used on imbalance data (Tharwat, 2020). One drawback of the method is that it does not distinguish between different values of the *Sensitivity* and *Specificity* when the *Youden's Index* is the same (Tharwat, 2020).

3.1.8 Mathews Correlation Coefficient

Matthews Correlation Coefficient was proposed by Brian W. Matthews in 1975 (Tharwat, 2020; Raschka, 2014; Chicco and Jurman, 2020). It has been used across many fields showing that it is an excellent metric (Chicco and Jurman, 2020). As stated by Tharwat (2020) it "represents the correlation between the observed and predicted classifications". By taking into account the number of *Positive* samples correctly labelled as well as the correctly labelled *Negative* samples, unlike other metrics (like F-score), it does not vary when classes are swapped (Chicco and Jurman, 2020). Canbek et al. (2017) recommend to take this metric for evaluating a method in addition to other more common metrics like *Accuracy*. However it is undefined if either a row or a column of the *Confusion Matrix* are zero (Chicco and Jurman, 2020). It is defined as follows (Luque et al., 2019; Tharwat, 2020; Powers, 2020; Chicco and Jurman, 2020):

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.14}$$

It is within the range $[-1, 1]$ (Luque et al., 2019; Tharwat, 2020; Canbek et al., 2017; Chicco and Jurman, 2020). A value of 1 denotes a perfect prediction of the ground truth, a value of -1 indicates that the prediction is as worse as it can be and a value of 0 illustrates that the prediction is no better than a random one (Tharwat, 2020; Raschka, 2014; Chicco and Jurman, 2020). If the method was able to detect most of the *Positive* samples and most of the *Negative* samples then the metric will have a high value close to 1 (Chicco and Jurman, 2020). When working with imbalance data it can be misleading as it is sensitive with this type of data (Luque et al., 2019; Tharwat, 2020). This metric can be normalized to lie within the range $[0, 1]$ as follows (Luque et al., 2019; Chicco and Jurman, 2020):

$$MCC_n = \frac{MCC + 1}{2} \tag{3.15}$$

Now $MCC_n = \frac{1}{2}$ represents the result of a random prediction (Chicco and Jurman, 2020).

Chapter 4

Simulations

In this chapter, interesting possible cases in bivariate space that can occur when trying to classify anomalies are going to be discussed. Also it will be helpful to get a better idea of the methods described above (Mahalanobis, LOF, DBSCAN, GMM Clustering and GMM Density threshold) by working with simple cases, in bivariate space where we can plot the points. The points will be simulated from bivariate normal distributions. For these simulations the package mvtnorm has been used in *R* (Genz et al., 2020). Furthermore code will be used for the methods (Mahalanobis, 1936; Fraley et al., 2012; Hahsler et al., 2017; Hu et al., 2015). The focus will be on two cases. The first case will have one cluster of normal points and one smaller cluster of anomalies. And the second case will have one cluster of normal points in the middle and two smaller cluster of anomalies on the sides. For each of the two cases, the performance of the methods will be evaluated with the metrics *Balanced Accuracy*,*Sensitivity* and *Specificity*, by varying the distance between the cluster means, the number of points in the non-anomalous cluster and the covariance matrices being equal or not. The Euclidean distance between the cluster means will vary from $2\sqrt{2}$ to $4\sqrt{2}$, the number of points will vary from 50 points to 100 points and the variances will vary from being equal to not being equal.

4.1 Case 1: One anomalous and one non-anomalous cluster

In this case there is a normal cluster and an anomalous cluster, the picture can be represented in Figure 4.1.

4.1.1 Mahalanobis distance

The results of the traditional Mahalanobis distance are presented in table 4.8. With respect to the Euclidean distance of the cluster means, the traditional Mahalanobis distance stays pretty constant for every metric when it changes from $4\sqrt{2}$ to $2\sqrt{2}$, however it tends to improve as this distance increases, specially when the number of points is small and the covariance matrices are different. Eventhough the *Specificity* stays pretty much the same, the *Sensitivity* drops when the number of normal points decreases, this can be caused by the influence of the anomalies on the parameters of

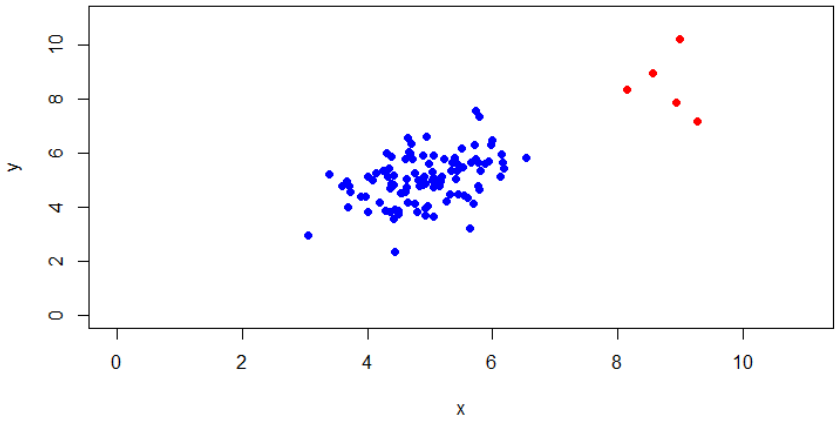

Figure 4.1: Plot of a normal cluster(blue) and an anomalous cluster (red)

the sample mean and covariance matrix, as in this case the proportion of anomalies has increased from 5% to 10%. When changing from having different covariances to equal covariances the *Specificity*, *Sensitivity* and *Balanced Accuracy* tends to improve. The performance of the more robust Mahalanobis method is better, specially on the *Sensitivity* metric, but it follows a similar pattern with respect to the number of points in the normal cluster, the distance between the means and the covariances being equal. However there are some cases that do not follow this pattern with respect to *Sensitivity* reaching the perfect score when the distance between the means is $2\sqrt{2}$, the number of normal points is 100 and the covariances are equal.

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous cluster	% Anomalies	Equal covariance matrices
0.8950	0.8000	0.9900	$4\sqrt{2}$	100	5	5%	YES
0.8800	0.8000	0.9600	$4\sqrt{2}$	100	5	5%	NO
0.8850	0.8000	0.9700	$2\sqrt{2}$	100	5	5%	YES
0.8800	0.8000	0.9600	$2\sqrt{2}$	100	5	5%	NO
0.7900	0.6000	0.9800	$4\sqrt{2}$	50	5	10%	YES
0.7900	0.6000	0.9800	$4\sqrt{2}$	50	5	10%	NO
0.8000	0.6000	1.0000	$2\sqrt{2}$	50	5	10%	YES
0.6900	0.4000	0.9800	$2\sqrt{2}$	50	5	10%	NO

Table 4.1: Results of the traditional Mahalanobis distance method when applied to the Case 1

4.1.2 LOF method

The results for the LOF method are summarized in Table 4.3. As recommended by (Breunig et al., 2000) a range of values for the parameter $MinPts$ is taken, for this reason, $MinPts$ is chosen to be between 10 and 50. Here, the assumption that there

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous cluster	% Anomalies	Equal covariance matrices
0.8950	0.8000	0.9900	$4\sqrt{2}$	100	5	5%	YES
0.8800	0.8000	0.9600	$4\sqrt{2}$	100	5	5%	NO
0.9800	1.0000	0.9600	$2\sqrt{2}$	100	5	5%	YES
0.8800	0.8000	0.9600	$2\sqrt{2}$	100	5	5%	NO
0.8800	0.8000	0.9600	$4\sqrt{2}$	50	5	10%	YES
0.7900	0.6000	0.9800	$4\sqrt{2}$	50	5	10%	NO
0.8000	0.6000	1.0000	$2\sqrt{2}$	50	5	10%	YES
0.8900	0.8000	0.9800	$2\sqrt{2}$	50	5	10%	NO

Table 4.2: Results of the more robust Mahalanobis distance method when applied to the Case 1

is no access to graphical representation is taken as the method will count with the advantage that by choosing the parameter to be greater than the largest anomalous cluster then the method will obtain perfect performance, hence the choice is not based on any previous graphical knowledge. The value of 10 is selected in order to avoid high statistical fluctuations between the LOF scores of points belonging to the same cluster. Since the anomalous clusters are considered to have less than 50 observations, this value is chosen to be the maximum of the range. Furthermore, the threshold value picked for this analysis is 1.6 due to the consideration that points with higher scores than this value will have higher chances of being anomalous. With these parameters chosen, the performance of the method is very good. As the Euclidean distance between the means increases from $2\sqrt{2}$ to $4\sqrt{2}$, the performance of the method improves, specially when the number of normal points decreases. When the covariance matrices changes from being different to being equal there is a small improvement on the *Specificity* metric, however the *Sensitivity* remains pretty much the same, this is expected as the method is able to manage effectively data sets with clusters of different densities.

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous cluster	% Anomalies	Equal covariance matrices
0.9950	1.0000	0.9900	$4\sqrt{2}$	100	5	5%	YES
0.9700	1.0000	0.9400	$4\sqrt{2}$	100	5	5%	NO
0.9850	1.0000	0.9700	$2\sqrt{2}$	100	5	5%	YES
0.8700	0.8000	0.9400	$2\sqrt{2}$	100	5	5%	NO
0.9700	1.0000	0.9400	$4\sqrt{2}$	50	5	10%	YES
0.9500	1.0000	0.9000	$4\sqrt{2}$	50	5	10%	NO
0.8000	0.6000	1.0000	$2\sqrt{2}$	50	5	10%	YES
0.7900	0.6000	0.9800	$2\sqrt{2}$	50	5	10%	NO

Table 4.3: Results of the LOF method when applied to the Case 1

4.1.3 DBSCAN algorithm

It can be seen in Table 4.4 the results corresponding to DBSCAN. The parameters are selected following the recommendations of (Ester et al., 1996; Sander et al., 1998), that suggest to choose $MinPts = 4$ for bivariate data sets. And the parameter ϵ is calculated by the k-distance graph for each of the samples considered. The performance of the method in the detection of anomalies is magnificent, obtaining a perfect score for the *Sensitivity* in every possible case, this demonstrates the power of this method when the right choice of the parameters are taken. The *Specificity* is close to 1 in every case as well, except on the penultimate case where the *Specificity* drops considerably to 0.7800 which is surprising considering the value in the other cases. Although it was not explicitly designed to detect anomalies its performance is incredibly good, surpassing other methods like LOF and the two Mahalanobis methods that were designed to detect these anomalies.

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous cluster	% Anomalies	Equal covariance matrices
0.9900	1.0000	0.9800	$4\sqrt{2}$	100	5	5%	YES
0.9800	1.0000	0.9600	$4\sqrt{2}$	100	5	5%	NO
0.9650	1.0000	0.9300	$2\sqrt{2}$	100	5	5%	YES
0.9850	1.0000	0.9700	$2\sqrt{2}$	100	5	5%	NO
0.9800	1.0000	0.9600	$4\sqrt{2}$	50	5	10%	YES
0.9900	1.0000	0.9800	$4\sqrt{2}$	50	5	10%	NO
0.8900	1.0000	0.7800	$2\sqrt{2}$	50	5	10%	YES
0.9400	1.0000	0.8800	$2\sqrt{2}$	50	5	10%	NO

Table 4.4: Results of the DBSCAN method when applied to the Case 1

4.1.4 GMM methods

For Gaussian mixture models two methods are implemented: the GMM Clustering method and the GMM Density Threshold method, both explained in section 2.4.2. The number of clusters is set to be $K = 2$. In both, the EM algorithm has been used to estimate the mean and covariance metrics of each of the classes, the anomalous and the non-anomalous. In the GMM Clustering method no additional calculation is required, however in the GMM Density Threshold Method an outlier score must be calculated. The methods are summarized in Table 4.5 and Table 4.6. The GMM Clustering method nearly obtains a perfect score of 1 for the *Specificity* and is the method that achieves the best values for this metric in most of the cases with respect to the other methods. It also acquires high values for the *Sensitivity* being 1 for most of the cases. The performance metrics improve as the distance between the means increases and when the covariances are equal. On the other hand, as the number of points of the normal cluster increases the *Sensitivity* decreases.

The GMM Density threshold method has very high values for the *Sensitivity* metric, however its results with respect to *Specificity* are the worst out of all the methods. This implies that it detects too many normal points as anomalous which is not beneficial. It does not follow a clear pattern when analyzed with respect to the variation of the

distance between the cluster means, the number of points of the normal cluster and the covariance matrices.

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous cluster	% Anomalies	Equal covariance matrices
1.0000	1.0000	1.0000	$4\sqrt{2}$	100	5	5%	YES
0.9000	0.8000	1.0000	$4\sqrt{2}$	100	5	5%	NO
1.0000	1.0000	1.0000	$2\sqrt{2}$	100	5	5%	YES
0.8000	0.6000	1.0000	$2\sqrt{2}$	100	5	5%	NO
1.0000	1.0000	1.0000	$4\sqrt{2}$	50	5	10%	YES
1.0000	1.0000	1.0000	$4\sqrt{2}$	50	5	10%	NO
0.9700	1.0000	0.9400	$2\sqrt{2}$	50	5	10%	YES
0.8600	0.8000	0.9200	$2\sqrt{2}$	50	5	10%	NO

Table 4.5: Results of the GMM Clustering method when applied to the Case 1

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous cluster	% Anomalies	Equal covariance matrices
0.6650	1.0000	0.3300	$4\sqrt{2}$	100	5	5%	YES
0.8400	1.0000	0.6800	$4\sqrt{2}$	100	5	5%	NO
0.7150	0.8000	0.6300	$2\sqrt{2}$	100	5	5%	YES
0.8500	1.0000	0.7000	$2\sqrt{2}$	100	5	5%	NO
0.8000	1.0000	0.6000	$4\sqrt{2}$	50	5	10%	YES
0.7500	1.0000	0.5000	$4\sqrt{2}$	50	5	10%	NO
0.7900	1.0000	0.5800	$2\sqrt{2}$	50	5	10%	YES
0.8400	1.0000	0.6800	$2\sqrt{2}$	50	5	10%	NO

Table 4.6: Results of the traditional GMM Density Threshold method when applied to the Case 1

4.2 Case 2: Two anomalous and one non-anomalous cluster

In this case a normal cluster and two anomalous clusters are considered, the plot is presented in Figure 4.2. The normal cluster is located between the two anomalous clusters.

4.2.1 Mahalanobis distance

Both the traditional Mahalanobis method and the more robust Mahalanobis method mantain a high *Specificity* through every case presented in Table 4.7, reaching values close to 1. The *Sensitivity* improves when the distance between the means and the number of points in the normal cluster increase. Furthermore, the covariances being equal tends to increment the *Sensitivity* values. The more robust Mahalanobis methods outperforms the traditional Mahalanobis distance in most of the situations, as expected.

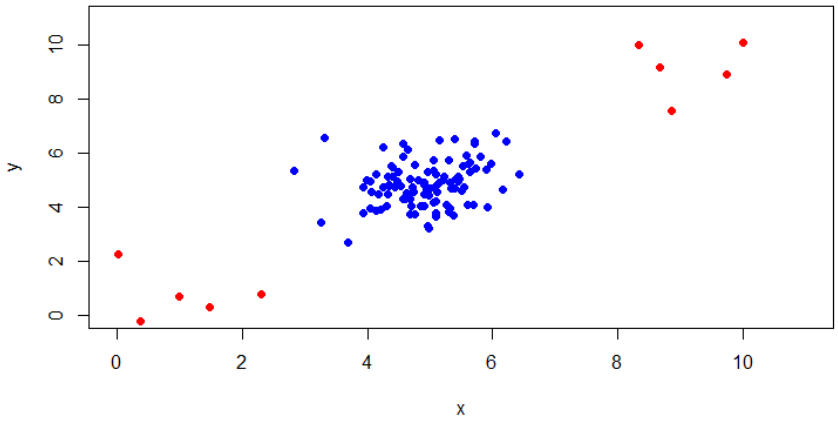

Figure 4.2: Plot of a normal cluster(blue) and two anomalous clusters (red)

It is also important to note that the performance of Mahalanobis when there are two anomalous clusters is worse than when there is only one anomalous cluster.

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous clusters	% Anomalies	Equal covariance matrices
0.9900	1.0000	0.9800	$4\sqrt{2}$	100	5	5%	YES
0.9400	0.9000	0.9800	$4\sqrt{2}$	100	5	5%	NO
0.7850	0.6000	0.9700	$2\sqrt{2}$	100	5	5%	YES
0.7900	0.6000	0.9800	$2\sqrt{2}$	100	5	5%	NO
0.7500	0.5000	1,0000	$4\sqrt{2}$	50	5	10%	YES
0.6900	0.4000	0.9800	$4\sqrt{2}$	50	5	10%	NO
0.6500	0.3000	1.0000	$2\sqrt{2}$	50	5	10%	YES
0.8000	0.6000	1.0000	$2\sqrt{2}$	50	5	10%	NO

Table 4.7: Results of the traditional Mahalanobis distance method when applied to the Case 2

4.2.2 LOF method

The LOF results are showed in Table 4.9, the parameter $MinPts$ is selected to be between the values of 10 and 50. Although the results when there are two anomalous clusters is worse than the simpler case when there is only one, the *Sensitivity* values achieve a perfect score when the distances between the clusters is $4\sqrt{2}$. The variation in the number of points of the normal cluster and the covariances being equal or different does not really affect the metrics.

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous clusters	% Anomalies	Equal covariance matrices
0.9900	1.0000	0.9800	$4\sqrt{2}$	100	5	5%	YES
0.9400	0.9000	0.9800	$4\sqrt{2}$	100	5	5%	NO
0.7850	0.6000	0.9700	$2\sqrt{2}$	100	5	5%	YES
0.7900	0.6000	0.9800	$2\sqrt{2}$	100	5	5%	NO
0.8000	0.6000	1.0000	$4\sqrt{2}$	50	5	10%	YES
0.7400	0.5000	0.9800	$4\sqrt{2}$	50	5	10%	NO
0.6400	0.3000	0.9800	$2\sqrt{2}$	50	5	10%	YES
0.8000	0.6000	1.0000	$2\sqrt{2}$	50	5	10%	NO

Table 4.8: Results of the more robust Mahalanobis distance method when applied to the Case 2

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous clusters	% Anomalies	Equal covariance matrices
0.9700	1.0000	0.9400	$4\sqrt{2}$	100	5	5%	YES
0.9900	1.0000	0.9800	$4\sqrt{2}$	100	5	5%	NO
0.7700	0.6000	0.9400	$2\sqrt{2}$	100	5	5%	YES
0.8750	0.8000	0.9500	$2\sqrt{2}$	100	5	5%	NO
0.9800	1.0000	0.9600	$4\sqrt{2}$	50	5	10%	YES
0.9700	1.0000	0.9400	$4\sqrt{2}$	50	5	10%	NO
0.8400	0.7000	0.9800	$2\sqrt{2}$	50	5	10%	YES
0.7900	0.6000	0.9800	$2\sqrt{2}$	50	5	10%	NO

Table 4.9: Results of the LOF method when applied to the Case 2

4.2.3 DBSCAN algorithm

The DBSCAN performance is displayed in Table 4.10. The parameters are chosen as follows $MinPts = 4$ and ϵ is is calculated by the k-distance graph for each of the samples considered, similar as when there was one anomalous cluster. It is able to obtain a perfect *Sensitivity* score in half of the cases, when the distance is $4\sqrt{2}$, similar to when there was only one cluster. However when a single cluster is present, the algorithm obtains a better performance than when there are two. The metrics tend to be independent to variations in the number of observations in the normal cluster. Varying the covariances of the clusters does not seem to affect the metrics either, this is not surprising as the method leads to misleading results only when there varying densities between the clusters (Khan et al., 2014; Mehrotra et al., 2017). The *Specificity* and *Balanced Accuracy* obtain very high values.

4.2.4 GMM methods

The GMM Clustering method gives an unexpected result when there is a normal point between two anomalous cluster. In this case the expected result is that each of the two estimated clusters contain one of the anomalous cluster and approximately half of the

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous clusters	% Anomalies	Equal covariance matrices
0.9800	1.0000	0.9600	$4\sqrt{2}$	100	5	5%	YES
0.9900	1.0000	0.9800	$4\sqrt{2}$	100	5	5%	NO
0.8250	0.7000	0.9500	$2\sqrt{2}$	100	5	5%	YES
0.8700	0.8000	0.9400	$2\sqrt{2}$	100	5	5%	NO
0.9100	1.0000	0.8200	$4\sqrt{2}$	50	5	10%	YES
0.9400	1.0000	0.8800	$4\sqrt{2}$	50	5	10%	NO
0.9000	0.9000	0.9000	$2\sqrt{2}$	50	5	10%	YES
0.9000	0.8000	1.0000	$2\sqrt{2}$	50	5	10%	NO

Table 4.10: Results of the DBSCAN method when applied to the Case 2

observations of the normal cluster. However, the algorithm at a large enough distance between the cluster means, is able to predict the two anomalous clusters as anomalies and its close to perfectly detecting the normal cluster. However, when the clusters come closer together the *Sensitivity* drops significantly. The *Specificity*, nevertheless maintains high values. Having equal covariances increases the metrics.

The GMM Density threshold, on the other hand obtains high values of *Sensitivity* at the expense of the *Specificity* being lower than for other methods. However the *Specificity* is higher when there are two anomalous clusters than when there is only one, which is surprising as the rest of the methods tend to acquire higher metric values for the second case as it is simpler.

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous clusters	% Anomalies	Equal covariance matrices
1.0000	1.0000	1.0000	$4\sqrt{2}$	100	5	5%	YES
0.9950	1.0000	0.9900	$4\sqrt{2}$	100	5	5%	NO
0.8400	0.7000	0.9800	$2\sqrt{2}$	100	5	5%	YES
0.6950	0.4000	0.9900	$2\sqrt{2}$	100	5	5%	NO
0.9800	1.0000	0.9600	$4\sqrt{2}$	50	5	10%	YES
0.9500	1.0000	0.9000	$4\sqrt{2}$	50	5	10%	NO
0.7000	0.4000	1.0000	$2\sqrt{2}$	50	5	10%	YES
0.6700	0.4000	0.9400	$2\sqrt{2}$	50	5	10%	NO

Table 4.11: Results of the GMM Clustering method when applied to the Case 2

BA	SENS	SPEC	Eucl. dist. cluster means	no. points normal cluster	no. points in anomalous clusters	% Anomalies	Equal covariance matrices
0.8400	1.0000	0.6800	$4\sqrt{2}$	100	5	5%	YES
0.8550	1.0000	0.7100	$4\sqrt{2}$	100	5	5%	NO
0.85000	0.8000	0.9000	$2\sqrt{2}$	100	5	5%	YES
0.8250	0.9000	0.7500	$2\sqrt{2}$	100	5	5%	NO
0.7600	1.0000	0.5200	$4\sqrt{2}$	50	5	10%	YES
0.7700	1.0000	0.5400	$4\sqrt{2}$	50	5	10%	NO
0.8200	1.0000	0.6400	$2\sqrt{2}$	50	5	10%	YES
0.8100	1.0000	0.6200	$2\sqrt{2}$	50	5	10%	NO

Table 4.12: Results of the GMM Density Threshold method when applied to the Case 2

Chapter 5

Results

The *Credit Card Fraud Detection* data set contains the transactions made by credit card users in September 2013. It is available in the following link *https* : *//www.kaggle.com/mlg − ulb/creditcardfraud*. The dimensionality of the data is 29 and it contains $284,807$ observations in total, with 0.1% of anomalies. The features of the data matrix are obtained after applying PCA but due to confidentiality issues the original features are not available. The only known variable that we will use is the transaction amount. The aim of this study is to compare the performance of different methods, and for that reason the dataset includes the true labels (fraud or normal) of each of the points. This labels will be excluded before applying the algorithms and then the estimated and true labels will be compared. These methods are implemented in R (Fraley et al., 2012; Hahsler et al., 2017; Hu et al., 2015; Mahalanobis, 1936).

From the *Credit Card Fraud Detection* database, a sample of 984 observations is taken, each time varying the percentage of anomalies taken from the 984 observations. The different methods are applied to this samples and the metrics of *Accuracy*, *Balanced Accuracy*, *Sensitivity*, *Specificity*, *Positive Predictive Value*, *Negative Predictive Value*, *Youden Index*, *Geometric Mean*, *Matthews Correlation Coefficient* and F_1*-score*.

5.1 Mahalanobis distance

The result of applying the two Mahalanobis distance methods explained in Chapter 2 are summarized in Table 1 and Table 2. The traditional Mahalanobis method is displayed in Table 1 and the one using more robust estimators for the mean and covariance matrix in Table 2. Both are effective methods specially when the percentage of anomalies is low, as the sensitivity is very high for this cases, reaching 1 when the percentage of anomalies is 0.002 and 0.001. However for percentages of anomalies higher than 10% the Sensitivity has low values, and hence the method misses many anomalies. Nevertheless, in these cases the Mahanobis method that uses more robust estimators obtains a better performance in Sensitivity than the traditional method. The performance of the Mahalanobis distance using robust estimators is at least as good as the one of the traditional Mahalanobis distance.

It is also important to highlight the high Specificity values that it maintains when varying the percentage of anomalies and as the Sensitivity increases. It is approximately

0.9 for each percentage of anomalies, as a result very few normal points will be wrongly estimated as anomalies. The Accuracy and Balance Accuracy increase as the percentage of anomalies decreases, this is expected as the number of anomalies correctly detected increases as the ratio of anomalies decreases.

It is relevant to see that the Positive Predictive Value decreases reaching to values close to 0 when the percentage of anomalies is close to 0%, so the ratio between correctly detected anomalies and detected anomalies decreases, however Sensitivity increases as anomalies decrease. This is because although more anomalies are detected correctly, the method also detects more normal points as anomalies, hence the denominator of the Positive Predictive Value increases (more normal points detected as anomalies) while the numerator decreases (the number of anomalies decreases). Furthermore, the Geometric Mean and the Youden Index increase with the decrease in the number of anomalies, as this depends on the Sensitivity and Specificity, and this are high when the number of anomalies is low.

Also, the F-measure is low for low percentage of anomalies as it depends on the Positive Predictive Value, that as we discussed above it decreases. Finally Matthews Correlation Coefficient tends to be stable, however it takes the lowest values when the proportion of anomalies is low. Nevertheless these two metrics together with the Positive Value tend to be very sensitive to imbalance data sets which is the case.

Ratio anom.	ACC	BA	SENS	SPEC	PPV	NPV	YI	GM	MCC_n	F_1
0.5	0.5528	0.5529	0.1931	0.9126	0.6884	0.5307	0.1057	0.4198	0.7610	0.3016
0.4	0.6565	0.5875	0.2411	0.9339	0.7090	0.6482	0.1750	0.4745	0.6250	0.3598
0.3	0.7439	0.6155	0.2949	0.9361	0.6641	0.7562	0.2310	0.5254	0.6558	0.4084
0.2	0.8161	0.6472	0.3655	0.9288	0.5625	0.8540	0.2943	0.5826	0.6751	0.4431
0.1	0.8831	0.8358	0.5510	0.9199	0.4320	0.9488	0.4709	0.7119	0.7117	0.4843
0.05	0.9085	0.8358	0.7551	0.9166	0.3217	0.9862	0.6717	0.8319	0.7274	0.4512
0.025	0.9167	0.8793	0.8400	0.9187	0.2121	0.9955	0.7193	0.8785	0.6985	0.3387
0.01	0.8913	0.8956	0.9000	0.8912	0.0782	0.9988	0.7912	0.8956	0.6235	0.1439
0.002	0,8994	0.9496	1.0000	0.8992	0.0198	1.0000	0.8992	0.9483	0.5667	0.0388
0.001	0.8994	0.9496	1.0000	0.8993	0.0100	1.0000	0.8993	0.9483	0.5464	0.0198

Table 5.1: Results of the Mahalanobis distance traditional method when applied to the *Credit Card Fraud data set*

Ratio anom.	ACC	BA	SENS	SPEC	PPV	NPV	YI	GM	MCC_n	F_1
0.5	0.5620	0.5620	0.2134	0.9106	0.7047	0.5365	0.1240	0.4408	0.5865	0.3276
0.4	0.6616	0.5972	0.2741	0.9203	0.6968	0.6550	0.1944	0.5022	0.6308	0.3934
0.3	0.7419	0.6218	0.3220	0.9216	0.6376	0.7605	0.2436	0.5448	0.6557	0.4279
0.2	0.8384	0.6840	0.4264	0.9416	0.6462	0.8677	0.3680	0.6336	0.7174	0.5138
0.1	0.8862	0.7462	0.5714	0.9210	0.4444	0.9511	0.4924	0.7254	0.7207	0.5000
0.05	0.9045	0.8240	0.7347	0.9134	0.3077	0.9850	0.6481	0.8192	0.7178	0.4337
0.025	0.9075	0.8552	0.8000	0.9103	0.1887	0.9943	0.7103	0.8534	0.6803	0.3054
0.01	0.8963	0.8982	0.9000	0.8963	0.0818	0.9989	0.7963	0.8981	0.6268	0.1500
0.002	0.8923	0.9460	1.0000	0.8921	0.0185	1.0000	0.8921	0.9445	0.5643	0.0363
0.001	0.8648	0.9324	1.0000	0.8647	0.0075	1.0000	0.8647	0.9299	0.5402	0.0149

Table 5.2: Results of the robust Mahalanobis distance method when applied to the *Credit Card Fraud data set*

5.2 Local Outlier Factor(LOF)

The results of the LOF method are displayed in Table 3. As recommended by (Breunig et al., 2000) a range of values for the parameter $MinPts$ is taken, for this reason, $MinPts$ is chosen to be between 10 and 50. Furthermore, the threshold value picked for this analysis is 1.6 due to the consideration that points with higher scores than this value will have higher chances of being anomalous.

The metrics of Accuracy, Balanced Accuracy, Sensitivity, Negative Predictive Value, Youden Index and Geometric Mean increase as the percentage of the anomalies decreases. The Specificity and the Matthews Correlation Coefficient stay quite balance across all the percentages of anomalies. On the other hand the Positive Predictive Value and the F_1-score decrease as the percentage of anomalies decreases reaching values close to 0, which is not beneficial. However it has to be taken into account that these two metrics are not robust to imbalance data sets and can lead to misleading results. And also we obtain very high Sensitivity values which leads to detecting most of the anomalies and reaching to perfect results when these anomalies are scarse. So in general the performance is quite good.

Ratio anom.	ACC	BA	SENS	SPEC	PPV	NPV	YI	GM	MCC_n	F_1
0.5	0.5549	0.5549	0.1443	0.9655	0.8068	0.5301	0.1098	0.3733	0.5962	0.2448
0.4	0.6413	0.5630	0.1701	0.9559	0.7204	0.6330	0.1260	0.4032	0.6055	0.2752
0.3	0.7409	0.6027	0.2576	0.9478	0.6786	0.7489	0.2054	0.4941	0.6482	0.3735
0.2	0.8394	0.6884	0.4365	0.9403	0.6466	0.8696	0.3768	0.6407	0.7205	0.5212
0.1	0.9126	0.8244	0.7143	0.9345	0.5469	0.9673	0.6488	0.8170	0.7888	0.6195
0.05	0.9268	0.8648	0.7959	0.9337	0.3861	0.9887	0.7296	0.8621	0.7615	0.5200
0.025	0.9400	0.8524	0.7600	0.9447	0.2639	0.9934	0.7047	0.8473	0.7129	0.3918
0.01	0.9421	0.9213	0.9000	0.9425	0.1385	0.9989	0.8425	0.9210	0.6701	0.2401
0.002	0.9350	0.9674	1.0000	0.9348	0.0303	1.0000	0.9348	0.9669	0.5842	0.0588
0.001	0.9400	0.9700	1.0000	0.9400	0.0167	1.0000	0.9400	0.9695	0.5626	0.0329

Table 5.3: Results of the LOF method when applied to the *Credit Card Fraud data set*

5.3 DBSCAN

It can be seen in Table 4 the results corresponding to DBSCAN. The parameters are selected following the recommendations of (Ester et al., 1996; Sander et al., 1998). The parameter $MinPts$ is chosen to be $MinPts = 2q = 58$. And the parameter ϵ is calculated by the k-distance graph for each of the samples considered.

The DBSCAN tends to give fairly constant results in most of the metrics considered across the different percentages of anomalies. It is one of the methods that obtains higher Positive Predictive Values across all the percentage of anomalies and reaches values of approximately 0.8 when the percentage of anomalies is high. This results in having large F_1-scores as well as this metric depends on the Positive Predictive Value.

Despite the fact that the percentage of anomalies is very high, the method achieves important results in the detection of anomalies because the Sensitivity drop below values of 0.8 approximately, in contrast with other methods that start with considerably lower Sensitivity values. However, the Specificity tends to fluctuate more but it is still

very high since it is greater than approximately 0.7 for every percentage of anomalies. Having values of Sensitivity and Specificity close to 1 derive to other metrics such as the Balanced Accuracy, the Geometric Mean and the Youden Index having high values.

Ratio anom.	ACC	BA	SENS	SPEC	PPV	NPV	YI	GM	MCC_n	F_1
0.5	0.8333	0.8333	0.8760	0.7907	0.8071	0.8644	0.6667	0.8544	0.8346	0.8401
0.4	0.8486	0.8425	0.8122	0.8729	0.8101	0.8744	0.6851	0.8420	0.8424	0.8111
0.3	0.8618	0.8548	0.8373	0.8723	0.7373	0.9260	0.7096	0.8546	0.8649	0.7841
0.2	0.8415	0.8533	0.8731	0.8335	0.5677	0.9633	0.7066	0.8531	0.8063	0.6880
0.1	0.7012	0.7978	0.9184	0.6772	0.2394	0.9868	0.5956	0.7886	0.6835	0.3798
0.05	0.9339	0.8686	0.7959	0.9412	0.4149	0.9888	0.7371	0.8655	0.7728	0.5455
0.025	0.7663	0.8217	0.8800	0.7633	0.0884	0.9959	0.6433	0.8196	0.6164	0.1607
0.01	0.9858	0.9433	0.9000	0.9867	0.4091	0.9990	0.8867	0.9424	0.8008	0.5625
0.002	0.9888	0.9944	1.0000	0.9888	0.1539	1.0000	0.9888	0.9944	0.6950	0.2667
0.001	0.9858	0.9929	1.0000	0.9858	0.0667	1.0000	0.9858	0.9929	0.6282	0.1251

Table 5.4: Results of the DBSCAN method when applied to the *Credit Card Fraud data set*

5.4 Gaussian Mixtures

For Gaussian mixture models two methods are implemented: the GMM Clustering method and the GMM Density Threshold method, both explained in Chapter 2. In both, the EM algorithm has been used to estimate the mean and covariance metrics of each of the classes, the anomalous and the non-anomalous. In the GMM Clustering method no additional calculation is required, however in the GMM Density Threshold Method an outlier score must be calculated.

GMM Clustering obtains better performance across the different percentages of anomalies than the GMM Density Threshold Method. Additionally, it is worth highlighting from GMM Clustering, its ability to distinguish between classes when the ratio of anomalies is very small, obtaining levels of Sensitivity and Specificity close to 1. This method presents a very good balance between these two metrics, that is present in the values of the Youden Index, Balanced Accuracy and Geometric Mean metrics, specially for small percentages of anomalies. Finally, when there is a high number of anomalies, the method manages to maintain not only a high value for Sensitivity but also for the Positive Predictive Value high which means out of all the anomalies that it predicts, most of them are correctly predicted reaching levels of approximately 0.8.

With regard to GMM Density Threshold method, it should be stressed the low Sensitivity that approaches 0 when the ratio of anomalies is very high, which may be due to the fact that as the number of anomalies increases the low-density regions where these points tend to be located decrease and the method performs poorly in these cases. However, when the percentage of abnormalities decreases, the sensitivity becomes higher. The specificity maintains extremely high levels across the ratio of anomalies, despite having such low sensitivity. It is the method that obtains the worst results in most metrics and the least recommended to use if the percentage of anomalies is not known.

Ratio anom.	ACC	BA	SENS	SPEC	PPV	NPV	YI	GM	MCC_n	F_1
0.5	0.7378	0.7378	0.6260	0.8496	0.8063	0.6944	0.4756	0.7293	0.7440	0.7048
0.4	0.7368	0.7244	0.6624	0.7864	0.6744	0.7772	0.4488	0.7217	0.7252	0.6383
0.3	0.8923	0.8766	0.8373	0.9158	0.8098	0.9293	0.7531	0.8757	0.8731	0.8233
0.2	0.8181	0.7949	0.7563	0.8335	0.5321	0.9318	0.5898	0.7940	0.7616	0.6247
0.1	0.8720	0.8926	0.9184	0.8668	0.4327	0.9897	0.7852	0.8922	0.7880	0.5882
0.05	0.8679	0.8918	0.9184	0.8652	0.2636	0.9951	0.7836	0.8914	0.7249	0.4092
0.025	0.8730	0.8569	0.8400	0.8738	0.1479	0.9953	0.7138	0.8567	0.6598	0.2515
0.01	0.9248	0.9620	1.0000	0.9240	0.1191	1.0000	0.9240	0.9612	0.6659	0.2128
0.002	0.9451	0.9725	1.0000	0.9450	0.0357	1.0000	0.9450	0.9721	0.5662	0.0689
0.001	0.9553	0.9776	1.0000	0.9552	0.0222	1.0000	0.9552	0.9773	0.5729	0.0434

Table 5.5: Results of the GMM Clustering method when applied to the *Credit Card Fraud data set*

Ratio anom.	ACC	BA	SENS	SPEC	PPV	NPV	YI	GM	MCC_n	F_1
0.5	0.4909	0.4909	0.0142	0.9675	0.3043	0.4953	0.0018	0.1172	0.4698	0.0271
0.4	0.7175	0.7092	0.6675	0.7508	0.6415	0.7718	0.4183	0.7079	0.7079	0.6542
0.3	0.7043	0.5145	0.0407	0.9884	0.6000	0.7064	0.0291	0.2006	0.5472	0.0762
0.2	0.7896	0.5108	0.0457	0.9759	0.3214	0.8033	0.0216	0.2112	0.5255	0.0800
0.1	0.8293	0.8734	0.9286	0.8183	0.3611	0.9904	0.7469	0.8717	0.7562	0.5200
0.05	0.7967	0.8544	0.9184	0.7904	0.1867	0.9946	0.7088	0.8520	0.6793	0.3103
0.025	0.8354	0.8571	0.8800	0.8342	0.1216	0.9963	0.7142	0.8568	0.6451	0.2137
0.01	0.9400	0.9697	1.0000	0.9394	0.1449	1.0000	0.9394	0.9692	0.6845	0.2531
0.002	0.8933	0.9465	1.0000	0.8931	0.0187	1.0000	0.8931	0.9450	0.5646	0.0367
0.001	0.9511	0.9756	1.0000	0.9512	0.0204	1.0000	0.9512	0.9753	0.5697	0.0400

Table 5.6: Results of the GMM Density Threshold method when applied to the *Credit Card Fraud data set*

5.5 Summary

In this section a series of comparative figures have been incorporated to analyse the performance of the methods with respect to the metrics of *Sensitivity*, *Specificity*, *Balanced Accuracy* and F_1-*measure* when the ratio of anomalies it is progressively increased.

5.5.1 Sensitivity

Figure 5.1 illustrates that every method tends to improve its Sensitivity when the percentage of anomalies decreases, this is not surprising as nearly every method except GMM Clustering was designed to find anomalies, and hence they will discriminate better this anomalies when their proportion in the data set is smaller. When the percentage of anomalies is low they have Sensitivity very close to 1, for these kinds of ratios the metric for the methods is very similar, however as the percentage of anomalies increases the method's Sensitivities tend to diverge to different values. The method with the worst performance in *Sensitivity* is the GMM Density Threshold having values close to 0 for some cases. On the other hand the GMM Clustering and DBSCAN methods are more stable across the different percentage of anomalies. The two Mahalanobis methods and the LOF obtain similar results.

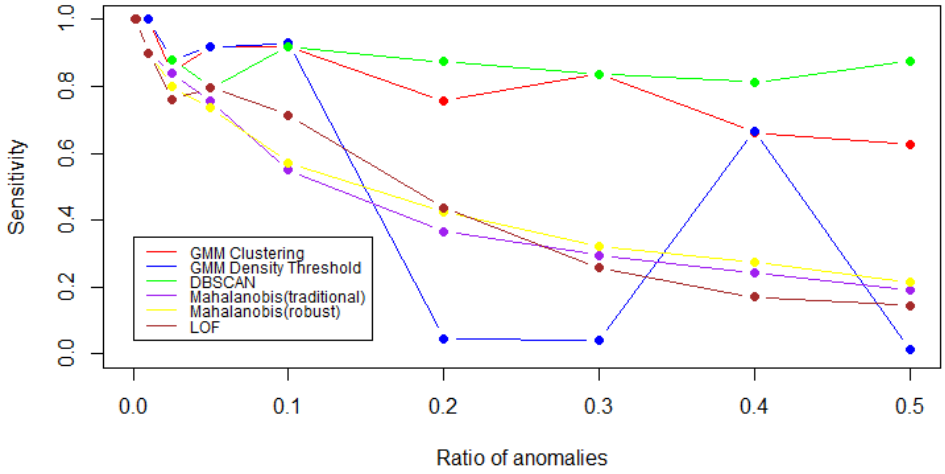

Figure 5.1: Plot of the *Sensitivity* metric against the ratio of anomalies for the different methods

5.5.2 Specificity

The *Specificity* reaches high values for every method, as can be observed in Figure 5.2. The methods that maintain higher values across the different ratios of anomalies are the LOF algorithm and the traditional Mahalanobis distance method. For this reason these two methods are the most suitable for correctly detecting normal points. DBSCAN obtains the lowest *Specificity* value out of all the methods and together with GMM Density Threshold are the ones that present greater fluctuation in their *Specificity* values.

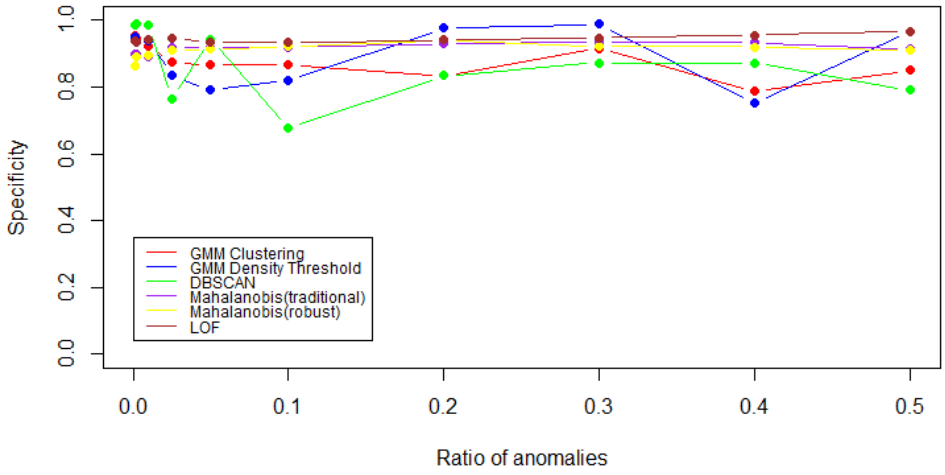

Figure 5.2: Plot of the *Specificity* metric against the ratio of anomalies for the different methods

5.5.3 Balanced Accuracy

Considering that the *Balanced Accuracy* is the mean between the *Sensitivity* and *Specificity* (Canbek et al., 2017; Tharwat, 2020), it depends on both of these metrics. In Figure 5.3 the results for the *Balanced Accuracy* are diplayed. For every method the *Balanced Accuracy* increases as the ratio of anomalies decreases. DBSCAN presents the better results as it obtains high and stable values for this metric. Conversely, GMM Density Threshold has the worst performance, this is expected as it was one of the worst methods for both the *Specificity* and the *Sensitivity*. LOF and both Mahalanobis methods achieve similar results with regard to this metric.

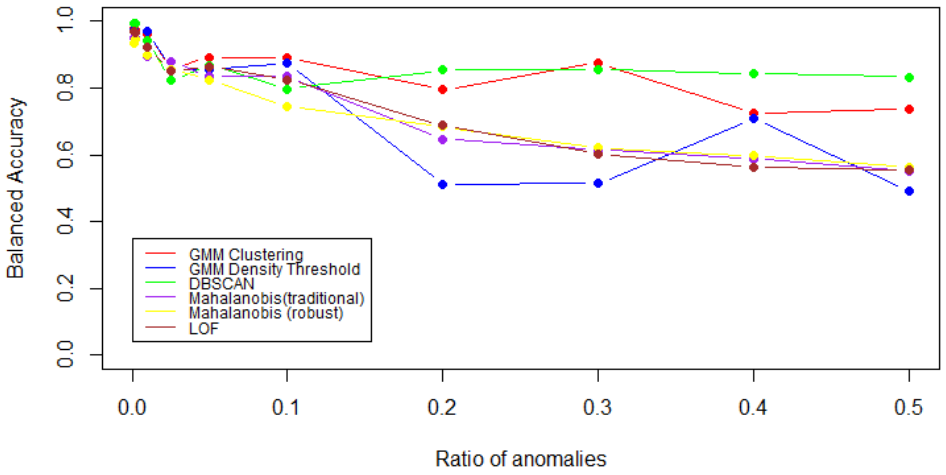

Figure 5.3: Plot of the *Balanced Accuracy* metric against the ratio of anomalies for the different methods

5.5.4 F_1-score

The F_1-*score* combines the metrics *Precision* and *Sensitivity* (Raschka, 2014). This score tends to decrease as the ratio of anomalies decrease. This is due to the fact that the *Prediction* values are close to 0 for small ratios of anomalies. However the *Sensitivity* is close to 1 for these ratios. The F_1-*score* metric is not robust with respect to imbalanced data sets and needs to be analyzed carefully (Luque et al., 2019; Powers, 2015; Tharwat, 2020). DBSCAN once again has the best performance, together with the GMM Clustering method. On the other hand, the GMM Density Threshold method obtains the worst results. Additionally, the two Mahalanobis distance methods and the LOF method acquire similar results.

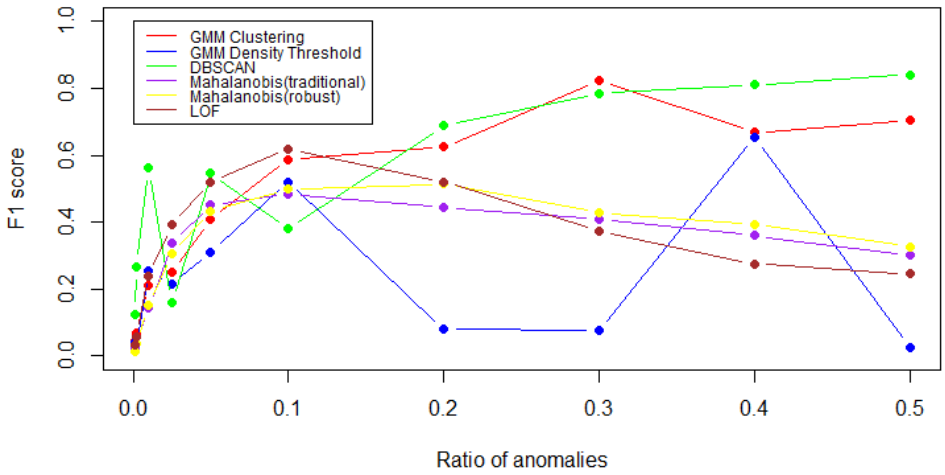

Figure 5.4: Plot of the F_1-*score* metric against the ratio of anomalies for the different methods

Chapter 6

Conclusion

Anomaly detection is becoming a very important statistical technique in certain applications such as the detection of credit card fraud, the object of this study. The main aim of this project is to evaluate and compare four unsupervised algorithms in detail, Mahalanobis distance, Local Outlier Factor (LOF), Density based spatial clustering of applications with noise (DBSCAN) and Gaussian Mixture Models (GMM), trying to analyse their performance in simulated and real data (*Credit Card Fraud data set*).

From the simulations it can be analysed that when there are anomalous clusters in the data set, the methods tend to improve their performance with regard to the metrics when at least one of the following situations is given: the distance between the means of the anomalous clusters and normal clusters increase, the number of points of the normal clusters increases and the clusters have equal variances. However LOF and DBSCAN achieve similar high results when some of these cases are not given, indicating their robustness in data sets with these characteristics, specially when the right value for their parameters is given.

Beyond the simulations, the performance of the different algorithms in the real data shows remarkable results. The GMM Density Threshold method obtains poor performance across most of the metrics considered, however the GMM Clustering achieves outstanding results despite being an algorithm that it is not specifically designed to detect anomalies. Something similar happens with DBSCAN, as although its main application is to discover clusters and isolate them from the anomalous points it is the one that maintains constant high results across the different percentages of anomalies. This can indicate that clustering and anomaly detection are closely related. Even though the Mahalanobis is a simple method it is able to compete with other more complex methods and being able to outperform them in some cases.

For further research it would be interesting to consider other distance metrics for the DBSCAN method as it is able to achieve excellent results with the Euclidean distance, hence it might achieve better results when other distance metrics are applied, like the Mahalanobis distance. Also, a study on how the performance of DBSCAN changes as the parameter value of $MinPts$ varies would be exciting, as it tends to be sensitive to the choice of $MinPts$.

Bibliography

Aggarwal, C. C. (2015). Outlier analysis. In *Data mining*, pp. 237–263. Springer.

Becker, C. and U. Gather (1999). The masking breakdown point of multivariate outlier identification rules. *Journal of the American Statistical Association 94*(447), 947–955.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Breunig, M. M., H.-P. Kriegel, R. T. Ng, and J. Sander (1999). Optics-of: Identifying local outliers. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 262–270. Springer.

Breunig, M. M., H.-P. Kriegel, R. T. Ng, and J. Sander (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104.

Campello, R. J., P. Kröger, J. Sander, and A. Zimek (2020). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10*(2), e1343.

Canbek, G., S. Sagiroglu, T. T. Temizel, and N. Baykal (2017). Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights. In *2017 International Conference on Computer Science and Engineering (UBMK)*, pp. 821–826. IEEE.

Çelik, M., F. Dadaşer-Çelik, and A. Ş. Dokuz (2011). Anomaly detection in temperature data using dbscan algorithm. In *2011 international symposium on innovations in intelligent systems and applications*, pp. 91–95. IEEE.

Chicco, D. and G. Jurman (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics 21*(1), 1–13.

De Maesschalck, R., D. Jouan-Rimbaud, and D. L. Massart (2000). The mahalanobis distance. *Chemometrics and intelligent laboratory systems 50*(1), 1–18.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological) 39*(1), 1–22.

Duan, L., L. Xu, F. Guo, J. Lee, and B. Yan (2007). A local-density based spatial clustering algorithm with noise. *Information systems 32*(7), 978–986.

Emmott, A., S. Das, T. Dietterich, A. Fern, and W.-K. Wong (2015). A meta-analysis of the anomaly detection problem. *arXiv preprint arXiv:1503.01158*.

Erar, B. (2011). Mixture model cluster analysis under different covariance structures using information complexity.

Ester, M., H.-P. Kriegel, J. Sander, X. Xu, et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, Volume 96, pp. 226–231.

Fluss, R., D. Faraggi, and B. Reiser (2005). Estimation of the youden index and its associated cutoff point. *Biometrical Journal: Journal of Mathematical Methods in Biosciences 47*(4), 458–472.

Fraley, C., A. Raftery, L. Scrucca, T. B. Murphy, M. Fop, and M. L. Scrucca (2012). Package 'mclust'.

Gan, J. and Y. Tao (2017). On the hardness and approximation of euclidean dbscan. *ACM Transactions on Database Systems (TODS) 42*(3), 1–45.

Genz, A., F. Bretz, T. Miwa, X. Mi, F. Leisch, F. Scheipl, B. Bornkamp, M. Maechler, T. Hothorn, and M. T. Hothorn (2020). Package 'mvtnorm'. *Journal of Computational and Graphical Statistics 11*, 950–971.

Ghorbani, H. (2019). Mahalanobis distance and its application for detecting multivariate outliers. *Facta Univ Ser Math Inform 34*, 583–95.

Goldstein, M. and S. Uchida (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one 11*(4), e0152173.

Hadi, A. S. (1992). Identifying multiple outliers in multivariate data. *Journal of the Royal Statistical Society: Series B (Methodological) 54*(3), 761–771.

Hahsler, M., M. Piekenbrock, S. Arya, D. Mount, and M. M. Hahsler (2017). Package 'dbscan'.

Han, J., M. Kamber, and J. Pei (2011). Data mining concepts and techniques third edition. *The Morgan Kaufmann Series in Data Management Systems 5*(4), 83–124.

Hawkins, D. M. (1980). *Identification of outliers*, Volume 11. Springer.

Healy, M. (1968). Multivariate normal plotting. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 17*(2), 157–161.

Hossin, M. and M. Sulaiman (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process 5*(2), 1.

Hu, Y., W. Murray, Y. Shan, and M. Y. Hu (2015). Package rlof. *Online] https://cran. r-project. org/web/packages/Rlof/Rlof. pdf*.

Im Walde, S. S. (2006). Experiments on the automatic induction of german semantic verb classes. *Computational Linguistics 32*(2), 159–194.

Khan, K., S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady (2014). Dbscan: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pp. 232–238. IEEE.

Kohl, M. (2012). Performance measures in binary classification. *International Journal of Statistics in Medical Research 1*(1), 79–81.

Kriegel, H.-P., P. Kröger, E. Schubert, and A. Zimek (2009). Loop: local outlier probabilities. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 1649–1652.

Krzanowski, W. (2000). *Principles of multivariate analysis*, Volume 23. OUP Oxford.

Lazarevic, A., L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava (2003). A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the 2003 SIAM international conference on data mining*, pp. 25–36. SIAM.

Lee, J., B. Kang, and S.-H. Kang (2011). Integrating independent component analysis and local outlier factor for plant-wide process monitoring. *Journal of Process Control 21*(7), 1011–1021.

Leys, C., O. Klein, Y. Dominicy, and C. Ley (2018). Detecting multivariate outliers: Use a robust variant of the mahalanobis distance. *Journal of Experimental Social Psychology 74*, 150–156.

Luque, A., A. Carrasco, A. Martín, and A. de las Heras (2019). The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition 91*, 216–231.

Mahalanobis, P. C. (1936). On the generalized distance in statistics. National Institute of Science of India.

McLachlan, G. J. (1999). Mahalanobis distance. *Resonance 4*(6), 20–26.

Mehrotra, K. G., C. K. Mohan, and H. Huang (2017). *Anomaly detection principles and algorithms*. Springer.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nguyen, H. (2015). Finite mixture models for regression problems.

Parikh, R., A. Mathai, S. Parikh, G. C. Sekhar, and R. Thomas (2008). Understanding and using sensitivity, specificity and predictive values. *Indian journal of ophthalmology 56*(1), 45.

Penny, K. I. (1996). Appropriate critical values when testing for a single multivariate outlier by using the mahalanobis distance. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 45*(1), 73–81.

Powers, D. M. (2015). What the f-measure doesn't measure: Features, flaws, fallacies and fixes. *arXiv preprint arXiv:1503.06410*.

Powers, D. M. (2020). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Raschka, S. (2014). An overview of general performance metrics of binary classifier systems. *arXiv preprint arXiv:1410.5330*.

Reddy, A., M. Ordway-West, M. Lee, M. Dugan, J. Whitney, R. Kahana, B. Ford, J. Muedsam, A. Henslee, and M. Rao (2017). Using gaussian mixture models to detect outliers in seasonal univariate network traffic. In *2017 IEEE Security and Privacy Workshops (SPW)*, pp. 229–234. IEEE.

Rencher, A. C. (2003). *Methods of multivariate analysis*, Volume 492. John Wiley & Sons.

Sander, J., M. Ester, H.-P. Kriegel, and X. Xu (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery 2*(2), 169–194.

Sawant, K. (2014). Adaptive methods for determining dbscan parameters. *International Journal of Innovative Science, Engineering & Technology 1*(4), 329–334.

Schubert, E., J. Sander, M. Ester, H. P. Kriegel, and X. Xu (2017). Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS) 42*(3), 1–21.

Sokolova, M., N. Japkowicz, and S. Szpakowicz (2006). Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pp. 1015–1021. Springer.

Stöckl, S. and M. Hanke (2014). Financial applications of the mahalanobis distance. *Applied Economics and Finance 1*(2), 78–84.

Tharwat, A. (2020). Classification assessment methods. *Applied Computing and Informatics*.

Xiang, S., F. Nie, and C. Zhang (2008). Learning a mahalanobis distance metric for data clustering and classification. *Pattern recognition 41*(12), 3600–3612.

Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer 3*(1), 32–35.

Acknowledgements

I would like to thank Professor Konstantinos Perrakis who supervised this project.